

OFFICE AUTOMATION SYSTEM : IMPLEMENTATION OF AN OFFICE LANGUAGE

**A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY**

**by
P. R NARAYANAN**

**to the
INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
AUGUST, 1985**

to

my parents

00
IIT KANDLA
CENTRAL LIBRARY
No. **A** 91879

IMEP-1905-M-NAR-OFF


11/6/85
Bm


ii

CERTIFICATE

This is to certify that the work titled, OFFICE AUTOMATION SYSTEM: IMPLEMENTATION OF AN OFFICE LANGUAGE by Shri P.R. Narayanan has been carried out under my supervision and has not been submitted elsewhere for a degree.

June, 1985


(S. Sadagopan)
Assistant Professor
Industrial and Management Engg. Program
Indian Institute of Technology
Kanpur 208 016

29/6/85


ACKNOWLEDGEMENTS

It is with immense pleasure and great respect that I express my deepest sense of gratitude to Dr. S. Sadagopan for his invaluable guidance and encouragement throughout the course of this work.

I take this opportunity to express my sincere thanks to all members of the IME family who made my stay at IIT Kanpur a memorable one.

I also wish to thank Swami Anand Chaitanya for his neat typing and Shri Buddhi Ram Kandiyal for his cyclostyling work.

P.R. Narayanan

CONTENTS

<u>Chapter</u>		<u>Page</u>
	ABSTRACT	vii
I.	INTRODUCTION	1
	1.1 Introduction	
	1.2 Computers in Office	1
	1.2.1 Office by Example	4
	1.2.2 Other systems	4
	1.3 Executive Work-Station	5
	1.4 Expert Systems	5
	1.5 Scope of the Thesis	6
	1.6 Structure of the Report	7
II.	OFFICE AUTOMATION	8
	2.1 What is Office Automation?	8
	2.2 Evolution of Office Automation	9
	2.3 Present State of Art	9
	2.4 Motivation for Office Automation	10
	2.5 Office Productivity	12
III.	OFFICE AND ITS CONCEPTUAL MODEL	14
	3.1 What is an Office?	14
	3.2 Need for an Office Model	16
	3.3 OAM and OSL	16
	3.4 Office Language	17
	3.5 Office Model	18
	3.6 Office Operations	19
	3.6.1 Function	19
	3.6.2 Procedures	21
	3.7 Work-Station	24

<u>Chapter</u>		<u>Page</u>
IV.	SYSTEM DESIGN AND FACILITIES	25
	4.1 Office Language	25
	4.2 Parser	26
	4.3 Lexical Analysis	27
	4.4 Semantic Specification	28
	4.5 Command Parsing	29
	4.6 OA System	29
	4.7 OA System Facilities	30
	4.7.1 Electronic Filing and Retrieval	31
	4.7.2 Electronic Mailing and Despatching	31
	4.7.3 Mail Box Management	33
	4.7.4 Tracing	34
	4.7.5 Noting	35
	4.7.6 Addresses of Other Offices	35
	4.7.7 Creating Letters	36
	4.7.8 Output Using Type/Print/Read	36
	4.7.9 Office Text Processor	36
	4.7.10 Data-base queries	37
	4.7.11 On-line Assistance	37
	4.7.12 Feed-back and Error Messages	38
V.	CONCLUSIONS AND RECOMMENDATIONS	40
	5.1 Conclusions	40
	5.2 Recommendations	41
	REFERENCES	43
	APPENDICES	

ABBREVIATIONS

DOCNO	Document number
INFO	Information
OA System	The Office Automation system that has been developed
OAM	Office Analysis Methodology
OFL	Office Language
OSL	Office Specification Language

ABSTRACT

The present work concerns itself with the development of an Office Automation system for an organisation in which there exists information interchange between its offices. An attempt has been made to automate the general office tasks such as Mailing, Filing, Acknowledgement, Letter Processing, Tracing of letters/files etc. A command language, the Office Language (OFL), to drive the Office Automation system has been developed for the purpose.

The Office Automation system has been implemented on the DEC-10 system at IIT Kanpur. The system is capable of performing the tasks specified through the Office Language.

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION:

In recent years we have been witnessing a rapid revolution in the world of computer technology. This technology is advancing so rapidly, that interactive computing in business, is becoming common-place. Thus the age of non-programmer professional i.e. the age of end-users is emerging. These users, although professional in their own field, have neither the time nor the motivation to learn a conventional programming language. Furthermore, the spectrum of end-users is rapidly widening. Although, a couple of years ago end-user systems tried to address secretaries and clerks, now the range includes executives, managers, engineers and perhaps soon housewives. A consequence of cheaper available hardware is that users are becoming more sophisticated in their applications and require more functions with better flexibility.

1.2 COMPUTERS IN OFFICE:

Computers are making a big in-road into today's offices. The cost of hardware is decreasing at the rate of 30 to 40% annually [8]. The cost of computer logic circuits is expected to drop by a factor of more than 20 during the next decade.

Computer memory cost is expected to drop by a factor of 170 during the same period. In contrast to this behaviour, personnel costs in terms of salaries are on the rise. This means that the pay-out ratio, that is, ratio of cost of technology vs. cost of people, is dropping. Whereas the pay-out ratio in 1970 for a fixed amount of processing power was 4800, and in 1980 it was 210, in 1990 it is expected to be in the 2 to 8 range [8]. Hence, it is reasoned that rather than hire people, it makes good economic sense to purchase more technology. In this way one can avoid the increasing of work-force by providing the existing work-force with tools to relieve them of many of their repetitive jobs.

As more and more office functions become automated through diverse products either offered or to be offered in the market, the user has to learn different software manuals and languages to interface with these products. Thus an end-user may have to use one manual for data-processing transactions, another for document handling, another for electronic mailing. But as we shall see later, all these office functions have been accomplished in one simple Office Automation System.

Whereas in 1970's and early 1980's operational areas dominated the use of computer cycles, end-users (office workers) will dominate the use of computer cycles by 1990. While centralised systems will still exist to meet operational requirements,

the shift to end-user computing means that much of the data and computing power will be distributed to the lowest levels of the organisation [8].

In an office, most data are stored electronically in data-bases. The way in which data is stored in a data-base requires an application specialist to act as an intermediary between the user and the data. This is because the type of models of data organisation used and the way in which data is accessed requires some degree of specialised knowledge and programming skill [13].

In order that the office workers be provided with means to directly access the data, we need an end-user oriented language with which users can describe their application to the system. The language should require minimum training to be learnt and yet be powerful to cover a reasonably large domain of office functions.

To meet this objective, the end-user facilities will have to move away from employing syntax and commands for manipulating data, to what has been termed as direct manipulation [8]. Direct manipulation involves physical rather than syntactic constructs for carrying out various operations.

Several systems like Office by Example, Apple Lisa systems, employ this paradigm for manipulating data through high level non-procedural languages.

1.2.1 Office By Example (OBE) [15]:

It is a two-dimensional programming language and system that attempts to mimic manual procedures of business and is a natural extension of Query by Example (QBE).

The language for OBE represents an attempt to combine and unify aspects of word processing, data processing, report writing, graphics and electronic filing/mailling. With such a language the end-users (secretaries, clerks and engineers) are able to specify and store complex OBE programs thus developing their own applications. This concept is illustrated in Fig. 1 of Appendix A.

In OBE, the fundamental data object consists of forms. The user manipulates data in the data-base directly through the forms. Thus the paradigm for manipulating data objects corresponds very closely to the way in which the physical objects would be manipulated.

1.2.2 Other Systems.

In the Apple Lisa system [8] and Xerox Star system, in addition to representing data objects directly on the screen for manipulation, one can represent other system facilities on the screen. These representations are usually in terms of icons that present a pictorial representation of the facility. For example, a filing cabinet can be represented as a drawing of a filing cabinet. The filing operation can be defined as

the action of physically moving a data object such as a form to the filing cabinet icon.

1.3 EXECUTIVE WORK-STATION:

With the emphasis being laid on end-users, individual work-stations will come into use. The **falling** costs of micro and minicomputer have facilitated this change. The computing power, will, hence be distributed. While each station may have a local word-processor and may have access to a local data base, it would be hooked on to the mainframe by which it can have access to the general data base. Ideally, we are speaking of an architecture where word-processing functions and access to personal databases can be carried out locally at the terminal site. Data processing functions and access to large data bases are executed at the host, which may connect to other large mainframes via a network as illustrated in Fig. 2 of Appendix A.

Eventually a technological office environment will be created giving managers on-line access to corporate financial data, customer account information, transaction processing and stock market economic reports etc.

1.4 EXPERT SYSTEMS:

Even after modernizing office technologies today's office systems are passive. The reason is that most of the knowledge of what needs to be done in an office, resides with

the office worker. The office worker knows what stages are required to accomplish some goal, but not the computer system. Thus current office systems provide only tools to support individual office steps. In addition to this, specific knowledge could be encoded in the system, to enhance its capability. This would free the office worker from repetitive and routine processing jobs. The systems would then be called as Expert Systems as they can definitely replace an expert.

The Odyssey system implemented at Xerox Palo Alto research centre uses knowledge about trip planning to assist an user in filling out a collection of electronic forms [8].

1.5 SCOPE OF THE THESIS:

In this thesis an attempt has been made to design and implement an Office Automation system to carry out simple tasks that are common to most offices. For example tasks such as mailing, filing, sending/receiving acknowledgement, have been automated. This level of task support is only of limited effectiveness in addressing the problem of Office productivity. An Automated Office Information System, that is, a computer based application system that supports an entire office procedure [5] is very much preferable. However, such systems are software specific and office specific. A major impediment to the efficient development of such systems is the lack of tools and methodologies that an analyst can use, to describe the operations

in the office, in the process of designing and implementing an Automated Office Information system for it.

1.6 STRUCTURE OF THE REPORT:

In this report stress has been given to the approach that was taken for the OA system development. Details of the implemented system are also given.

- | | |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Chapter I | Chapter I gives a broad idea of the increasing usage of computers in offices and the development of office systems. |
| Chapter II | Chapter II focusses on Office Automation System - its evolution, current state, motivation for the system, and its advantages. |
| Chapter III | Chapter III deals with the development of a conceptual model of Office and the concept of Procedures, Functions, Events, Steps and States. |
| Chapter IV | In this, we focus, on the how the system was designed and the facilities made available on the system. |
| Chapter V | In Chapter V, the final chapter, we have made some concluding remarks on the OA system developed, scope for further work, recommendations and finally the limitations of the system. |

CHAPTER II

OFFICE AUTOMATION

2.1 WHAT IS OFFICE AUTOMATION?

The term Office Automation is usually taken to mean the introduction of technology into an office to automate it, an analogy taken from the automation of factories. The basic concept is correct, but by no means it is complete. Simply placing a word-processor in the middle of a busy office does not constitute automation [6].

Generally in any office of an organisation a substantial portion of time, 50 - 70% is spent in document preparation, document handling, mailing, filing and in communication. Typically managers spend 66 - 80% of their time in oral communications [4]. Harvey Poppel [4] reported that 40% of a typical manager's time is spent in mail processing, telephone calls and in business travel. The general business functions include typing, document preparation, mailing, filing, rapid publication of technical reports and organisation of conferences and symposia.

Mechanisation of these low level tasks may be characterised as activities in the domain of Office Automation. Interactive tools are provided to office-workers in the areas in which they are working.

2.2 EVOLUTION OF OFFICE AUTOMATION

Over the past few years we have seen a dramatic increase in the number of new products aimed at simplifying the office tasks such as writing memos/letters. Under the banner of word-processing and office of future, businesses are experimenting with long evolutionary processes to streamline information exchange in the office [2].

Twenty-five years ago a letter was written by hand or dictated to one's secretary who translated it later from ~~her~~ own shorthand. A draft was typed, retyped with several carbon copies. In time, the dictation equipment eliminated the need for shorthand skills. This was followed by photocopiers, which eliminated the need for carbon copies. In today's environment, the ability to store the text on a removable medium (card, cassette, floppy disc.) is a natural answer to the problem of retyping the document.

2.3 PRESENT STATE OF ART:

Before the 1980's there were three traditional technologies in the office which were relatively separate and independent. Data processing technology included computers, storage devices and output devices. Office technology included typewriters, copiers, adding and dictation machines. Communications technology included telephones, private branch exchanges (PBX's), telex machines etc. These technologies are converging to the

point that it is becoming impossible to view them as distinct technologies. The computer and communications technologies have converged to produce distributed data-processing, intelligent switching devices etc. The computer and office technologies have produced the computer, computer input microform (CIM), computer output microform (COM). The office and communications technologies have resulted in dial dictation, facsimile transmission, teleconferencing. All three technologies have combined to produce computer based messaging systems, portable intelligent terminals, communicating word processors, compound document storage (data, voice, text, image). The convergence is depicted in Fig. 3 of Appendix A.

Thus, the current state of art offering includes satellite transmission facilities, video teleconferencing, speech-filing systems and electronic fund transfer systems.

The number of companies wielding office equipment has grown greater than 1600; the number of software companies is in excess of 4000. Work-station functionality and user interfaces continue to improve and we are finally gaining experience with large office systems.

2.4 MOTIVATION FOR OFFICE AUTOMATION:

The momentum towards Office Automation is rooted in recent significant technological developments which have occurred,

simultaneously with an acute need to augment the productivity of the white collar worker. That is, a 'technology push' and a 'demand pull' provide the driving forces for the changes we are experiencing [4].

The powerful basic reason for automating the offices is that white collar salaries are a huge and intractable cost of doing business. In 1980, 60 of 1.3 trillion\$ paid out for wages, salaries, benefits in the U.S. went to the office workers [11].

The growth in productivity of white collar workers (6 - 10%) is very low when compared to the figures for farm workers (60%) and for blue-collar workers (30%). Considering the fact that 80-95% of their time is spent in communicating and managing information, any improvement in the communication process has enormous potential leverage for operating expense control [11].

At the other end of white collar workers is the secretary, 70 - 75% of whose time is spent in document handling (typing, copying dictation) and in taking telephone calls and messages. Much of the office automation is focussed on improvement of secretarial text processing and typing. There is also a sincere desire to eliminate the tedium associated with a secretary's job [2].

The environment businesses are experiencing is one of increasing complexity. Organisational success in these complex times is dependent on improving the quality of information made available to the management so that better business decisions may be made. This means information has to be more accurate and timely.

Insofar as the corporate motivation is concerned it becomes convenient to characterise their goals as falling into two areas [11].

(i) Cost displacement systems: which focus on the administrative parts of business, causing displacements usually in clerical/secretarial area.

(ii) Value added systems: which have the potential to free up managers and professionals time to undertake new and partially more creative work.

Integration of the two systems provide both support for clerical and administrative as well as managerial processes with the sought after benefits of both cost savings and increased managerial effectiveness.

2.5 OFFICE PRODUCTIVITY:

Being productive means 'producing or tending to produce goods or services having exchange value' [8]. The primary goal of Office Automation is the use of technology to improve productivity of the office.

The products of office work can be improved both in terms of their quantity (efficiency) and quality (effectiveness) through the use of cost displacement and value-added systems respectively.

Increased effectiveness occurs when the outputs that receivers and managers want are judged to be right. Office Automation contributes to effectiveness because it generates the capacity within which tasks can be performed to a higher degree of acceptance.

In an office environment the means of production are not as tightly controlled as in industrial environment. Inputs are not always standardised or of uniform quality. People have a great deal more discretion about how the final product-information will be produced. Productivity, thus can be improved by controlling the quality of work, streamlining the means of production so that more could be produced per unit time (increase in efficiency).

In addition to increasing productivity, Office Automation is intended to:

- keep the morale of office workers high
- give better services to customers
- do office work better rather than just faster.

CHAPTER III

OFFICE AND ITS CONCEPTUAL MODEL

3.1 WHAT IS AN OFFICE ?

There are a number of prevailing views of the office. This essentially depends on the type of approach. It is important to understand the main types of frame works so that the implementor of the office system can adopt or synthesise an appropriate approach for his/her organisation. Five approaches are reviewed below. [4].

1. Organisational and Communication Approaches:

These approaches view the organisation as a communication system. The objective of an office system, for those holding this view is to improve communication resulting in time savings, better access to information and improved organisational relationships.

2. Functional Approaches:

Functional approaches emphasise the impact of office systems on the underlying functions which the office exists to fulfil. To the functional practitioners, other approaches mistakenly confuse the ends (functions) that the office is realising with the means (e.g. communication) used to achieve them. To improve office functions, the designer should focus on improving office procedures using the new technology.

3. Information Resource Management Approaches:

Information system approaches go beyond the traditional MIS approaches to view office systems as enabling effective management of the information resource. Like money, people and facilities, information should be viewed as a resource which can be better managed through the use of new systems.

4. Decision Support System Approaches:

These approaches situate office system design and measurement in the concept of supporting the judgement of managers and other workers who make decisions. The objective of office systems is then to study effects of systems on the performance of complex unstructured or semi-structured activities.

5. Quality of Work Life Approaches:

Quality of work life approaches emphasise the impact of office systems on the nature of work and motivation of worker. Office system research and design and implementation may be done to jointly optimise the effectiveness of both the social and technical components of any work system.

For our purpose, the approach we take is a combination of approaches 1, 3 and 5.

Raymond R. Panko and Ralph Sprague Jr. of University of Hawaii [11] have classified the offices into two types, (i) Type I and (ii) Type II.

(i) Type I offices handle the organisation's high volume but routine information processing chores. Examples are accounting, payroll, billing, reservation etc.

(ii) Type II offices in contrast, are responsible for broad policy setting and professional functions in the firm.

Type II offices are, of course, a much more complex entity to address through Office Automation. In our present work we have tried to automate a conceptual office, whose activities are similar to that of Type I offices.

3.2 NEED FOR AN OFFICE MODEL:

Before we automate an office we need to be able to understand what goes on in an office. Thus we need to develop and use models that allow us to capture data processing and communication requirements in an office.

Once an office application is specified by a precise model it is much more amenable for computerisation. The models will allow easier development and maintenance of office applications. The models may also help in restructuring the existing workflow in an office, thus improving the productivity.

3.3 OAM AND OSL:

OAM (Office Analysis Methodology)[6] and OSL (Office Specification Language) [5] are two tools available for the system analyst to model office operations.

3.3.1 OAM:

OAM is designed to gather information about the operations of an office. It does not explicitly gather information about areas of office work that are badly in need of change nor does it identify leverage points. OAM study helps the analyst to have a good idea as to what the problem and opportunity areas are. This information is not collected in any specific way, rather it is something that the analyst runs into in the course of collecting the information that actually is specified.

3.3.2 OSL:

Significant difficulties in analysing and describing offices arise from the apparent complexity of office operations. A formal specification language can alleviate these problems in several ways. Firstly the formal nature of the language serves to eliminate ambiguities inherent in an English description of an office. Secondly the structure of language itself serves as a conceptual framework for thinking about and describing office work [5]. The major goals of OSL are thus to provide both a perspective with which to approach office analysis and a formal notation in which to communicate effectively the information elicited.

3.4 OFFICE LANGUAGE:

Office Language (OFL) has been developed using the LANG-PAK -an interactive language design system (explained in

more detail in Chapter IV). The commands to drive the OA system to do the required activities is given in OFL which resembles closely the English language.

3.5 OFFICE MODEL:

In terms of office data and procedures, models are of two types, those that emphasise data semantic aspects and those that emphasise communication aspects. In the data semantic approach to office modeling one tries to capture different types of data objects that exist and operations on those objects. In the latter model, one tries to model how the data objects are transmitted among the various work stations in the office and what processing each work-station in the office does on the data objects.

In the present work, an office model has been developed, based on the premise that an office has a mission, has control over some parts of its environment and has structure in terms of functions and procedures.

The office model can be described in terms of OBJECTS, FUNCTIONS and PROCEDURES. An OBJECT is an abstraction that may have associated with it one or more documents or other physical items and that is central to a series of activities. For e.g. in ordering for supplies, the object would be request for supplies, represented physically by a requisition form. A FUNCTION is the life history of an object consisting of atleast

three parts; the object's origination, management and its termination.

A function originates because of the recognition of a need. If leave is required, the letter writing function is invoked and the letter is written. Then with the help of mailing function the letter is mailed to the concerned authority, and the reply is awaited. Once the reply is received that marks the end of a series of functions.

The PROCEDURE is the primary descriptive tool of the model. A PROCEDURE is defined as a set of structurally related activities to be executed in a certain manner so as to accomplish an office function [9]. A PROCEDURE is described in terms of EVENTS, STEPS and STATES. An EVENT is anything that triggers an action and it can be a receipt date or a non-receipt of a document. A STEP is everything that can be done without having to wait for another EVENT. When the STEP has been completed, the object enters a STATE to wait for the next EVENT.

3.6 OFFICE OPERATIONS:

3.6.1 Function:

A FUNCTION is characterised by management of a set of objects over time; its goal is in the maintenance of these objects called resources in this context. The actions involved include monitoring anticipated events and responding to unanticipated ones.

In our Office Automation system mailed objects (letter or file) are entered into the Mail Log and information about them is maintained.

Example: LETTER = LACADEMIC has FROM OFFICE: IME, TO OFFICE = ME, ACK.RECD = NO. It means the letter LACADEMIC was mailed to the office ME and no acknowledgement has been received for the same.

The happening of LETTER = LACADEMIC is an EVENT and the other fields contain information (resources) about the letter. Management of these information is a FUNCTION. As new objects are mailed, the information in the log book is updated. When acknowledgements are received the information is modified appropriately.

Function Specification example:

EX 1: FUNCTION TRACE

OBJECT: <FILE/LETTER>

EX 2: FUNCTION FILE

OBJECT: <FILE/LETTER/INFO>

IN FILE: <FILE>

WITH DOC NO: <document number>

EX 3: MAIL FUNCTION

OBJECT: <LETTER> OR <NOTICE>

DESTINATION: <OFFICE> OR <ALL OFFICES>

3.6.2 Procedures:

These describe the active processing of objects from an initial state to some final state. A state is a point in the procedure at which processing of object is suspended, pending an external event. The response to an event is a step, which describes the processing that can be accomplished before awaiting the next event.

A procedure, therefore, has a definite invocation and a definite termination.

For example consider the purchase of some material by a department. The procedure PURCHASE is invoked when a need for the material is acutely felt.

A request letter is made, to be approved by the Head of the Dept. Approval of the Budget committee is sought after next and then the approval of the Director. Once the permission is given to the department, the department needs to raise quotations. The bids are analysed then and a vendor is recommended to the Purchase section. If necessary, the purchase is expedited by sending reminders to the Purchase section. After the arrival of a letter from the Purchase section that the said material has been purchased, inspected and approved, a stores indent is placed by the dept. The material is then transferred from the accounts of the stores to the accounts of the department. With this the purchase is over.

Procedure Specification Example:

Procedure PURCHASE

Step 1: Mail request letter to the Head of Dept.

State 2: Wait for approval

Event 1: Approval of Head of Dept. received.

Step 2: Mail request letter to Budget Committee
together with the Head's approval.

State 2: Wait for approval

Event 2: Approval is received

Step 3: Mail request letter to the Director.

State 3: Wait for approval.

Event 3: Approval is received

Step

4.1: Raise quotations

4.2: Analyse bids

4.3: Recommend vendor to the Purchase section.

State 4: Wait for Purchase section to purchase,
inspect and approve.

Event 4: Letter from Purchase section is received.

Step 5: Place stores indent

Event 5: Material is received by the Dept.

(Final state is reached)

Thus we see that a procedure may consist of a number of steps. After a step is over, invariably the system is in a waiting state for an event to occur, so that further processing could be done.

3.6.2.1 Steps and States:

Step is a set of activities that is performed at a given stage of a procedure. Normally a step is terminated because no further processing can be done without the occurrence of an external event. When the object is awaiting an occurrence it is in a particular state; in order for an object to reach a state, it must have gone through the processing specified in the steps proceeding the state.

3.6.2.2 Activities:

The steps of procedure consists of activities which are basic constructs of the Office Language (OFL). Activities may be executed in parallel or in sequence or according to a certain specified function [9]. An activity in the Office Language is usually described by a <verb> followed by an <Object>. A destination may or may not be specified.

- OPEN - Opens an object (a file)
- TRACE - traces the whereabouts of an object
(a file/a letter)
- NOTE - makes a note in an object
(a letter/a particular document in a file)

MAIL - mails an object

(a letter/a notice) to a specific destination.

3.7 WORK-STATION:

A primary assumption is that Office Automation is useful in an environment where there is information interchange between offices. Each office is equipped with a work-station (Fig. 4 of Appendix A) housing the Office Automation system software. The work-stations are linked via a main-frame computer (Fig. 5 of Appendix A).

In our work we have considered each office/work-station to have an unique project-programmer number. All transactions take place between the different offices. Each office is capable of performing the same activities as other offices.

CHAPTER IV

SYSTEM DESIGN AND FACILITIES

4.1 OFFICE LANGUAGE (OFL):

Developing a suitable language interface between the end-users and an application program is an important consideration when interacting with a non-computing community. The OFL has been developed with this in mind.

The OFL has been developed using LANG-PAK [7] -- an interactive language design system. Both syntactic and semantic specifications have been built in the OFL. Thus the OFL has its own grammar and vocabulary. The vocabulary could be increased, if necessary, by having a session with the LANG-PAK. The grammar of the language is given in Appendix B.

For our applications, the one unified language in Fig. 1 of Appendix A is OFL. OFL forms the command language of our OA system. The commands to drive the OA system are given in pseudo-English imperative sentences. For example

. GET MAIL BOX

this command opens the mail box and checks for any mail

. MAIL LETTER LXXX TO <PPN>

this command mails the letter LXXX from the office to a specified office denoted by the PPN.

With the present vocabulary of the OFL, a number of office activities (discussed under System Facilities in this Chapter) can be performed by our OA system.

4.2 PARSER:

The parsing algorithm employed in LANG-PAK is a left-to-right, top-down, fast-back algorithm [1]. That is, it analyses and translates the left-most part of a symbol sequence first and proceeds to the right only when necessary. The parsing algorithm builds the syntax tree from the start type (root node) and works downward to the end nodes, hence the classification top-down.

The top-down parsing method is goal oriented, that is at each successive level in constructing the syntax tree the parse machine sets as its current goal either a language specification type, terminal word set, or a parse machine instruction. This goal may in turn involve finding other language specification types as subgoals. Each subgoal is analysed in the language specification type definition from left to right and if all subgoals are successful, the goal itself is realised. Successful recognition of a subgoal will be passed back to the goal of which this goal is the subgoal.

The term fast-back refers to the way that the parse machine processes alternatives in a repeating specification. On encountering a repeating specification, the parse machine sets the first component of the first alternative as its current goal. If the parse machine fails in attempting to recognise this current goal, it then sets the first component of the next alternative as its current goal. This process is repeated until no more alternatives exist in which case the repeating specification can no longer be iterated. If the parse machine succeeds in recognising the first component of an alternative, it proceeds to set the second component of the alternative as its current goal. Failure to recognise any component of an alternative other than the first component, causes the alternative to fail and hence the repeating specification can no longer be iterated.

4.3 LEXICAL ANALYSIS:

The lexical analysis identifies terminal words in an input symbol sequence. In this top-down parsing method the lexical analyser (LEXANL) is called by the parse machine (PARSER) to decide if the non-null symbol sequence which appears next in the input symbol sequence matches a member of the current terminal word set specified in the language specification type definition under consideration. If this match fails, the lexical analyser reports the failure to the parse machine, which

in turn handles this failure according to the rules of the top-down, fast-back algorithm. A successful match by the LEXANL, results in reporting success to the parse machine. Along with success indication the lexical analyser generates a translation element. A translation element consists of a sequence of integers which represents the terminal word successfully matched. This sequence of integers consists of a translation code which represents the type of terminal word matched, followed by the remainder of the symbol sequence's translation.

4.4 SEMANTIC SPECIFICATIONS:

The semantic aspect of a language is concerned with "meaning" that is, the interpretation of the sentence. One may consider a mapping from the language to the process of satisfying the language request, and this mapping is what is meant by the semantics of the language.

The semantic constraints have been embodied in the OFL, the language of the Office Automation system.

When the parse machine needs to apply a semantic specification during the course of parsing an input symbol sequence, it invokes the semantic machine. The parse machine parses the translated form of the semantic specification (as produced by the semantic compiler) to the semantic machine which executes the semantic specification.

4.5 COMMAND PARSING:

The LANG-PAK parser produces three types of tables.

The first is a character table which translates the physical key strokes on the work-station key-board to some unique integer numbers which can be recognised by the language design system. This unique system is the LANG-PAK internal symbol (number) code. This is performed by two FORTRAN routines (1) IGETNO: which translates the character into the internal number code, (11) IGETCH: which translates the internal number code back to characters to be used in the application program. The advantage is that in FORTRAN it is more convenient to manipulate with numbers than with characters.

The second table is the command table produced during the design of the OFL. The command table defines the acceptable commands (both syntactically and semantically correct). It is the translation of the grammar of OFL into the internal LANG-PAK symbol code.

The third table is the translation table, which provides a mapping between the operations and functions to execute them. This table contains the translation elements, by which the OA system software does the required activity.

4.6 CA SYSTEM:

The OA system has been developed and implemented on the DEC-10 system available at IIT Kanpur. The software for the

system has been written in FORTRAN language.

The DEC-10 system operates on a time sharing basis. About 32 workstations (or terminals) are hooked on to the mainframe computer. Each user is identified by a unique Project-Programmer Number (PPN). Hence the most convenient way of representation of offices in such a computing environment is by a PPN. Each office houses the OA system software in its area and is capable of performing the activities (which shall be discussed under system facilities in this chapter).

The OA system is driven by commands given in OFL. The tables necessary to drive the LANG-PAK parser when parsing statements exist in a file on an external medium. This table forms the front end of our OA system. The OA system takes as its input the table from the external medium. The system then reads the statements in the language from the users input device (terminal) and passes the statement along with the grammar into the LANG-PAK parser. This method of execution is justified because the language processing takes only a small amount of time relative to the time to perform the office activity. If the statement parses correctly, its translation is used by the OA system software to perform the desired action.

4.7 OA SYSTEM FACILITIES:

The different facilities available with the OA system are discussed below. The commands used in the examples are indicative of the syntax but may not be exact.

4.7.1 Electronic Filing and Retrieval:

Filing is the process of maintaining several distinct pieces of information under a particular head. In physical terms, the head is a file containing the pieces of information. This activity is quite common in most offices.

In our OA system one can file (i) a letter (ii) a file (either whole or part of it) (iii) information in a file. Every piece of information is known by a unique document number.

. FILE <INFO/LETTER/FILE> WITH DOCNO (...)

Thus every file is sequential file with the index as DOCNO.

Documents may be retrieved from the file in the following manner.

. GET ALL DOCNO

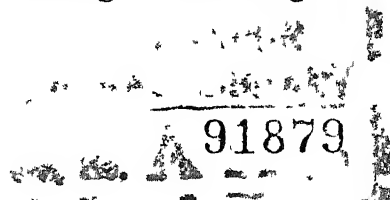
gives a list of all document numbers in a file

. GET COPY OF DOCNO (...,...,...,)

will get the copy of the given document numbers into a specified file from a specified file.

4.7.2 Electronic Mailing and Despatching:

Mailing and Despatching are functions that transfer information from office to a destination specified by the user. It may be through a letter, message (Mailing) or through a file (Despatching)



- . MAIL LETTER LXXX TO <destination>
- . DESPATCH FILE FXXX TO <destination>

Whenever a letter or file is sent to another office, an entry is made in the Mail log book of the sender. The Mail log book maintains information about objects that are sent. It is a relation with the fields FROM OFFICE, TO OFFICE, OBJECT NAME, ACKNOWLEDGEMENT RECD. (YES/NO). An entry is also made into the receivers Mail Box.

4.7.2.1 Receiving Mail:

The physical act of checking the Mail box for its contents is done by our OA system through

- . GET MAIL BOX

The object name and the senders office are displayed on the screen. The text of the material sent will be found in the Receiver's area (directory, for our purpose).

The OA system has been designed to send simultaneously an acknowledgement to the sender quoting the object name.

4.7.2.2 Receiving Acknowledgement:

One may get to know the acknowledgements one has received by typing

. GET ACKNOW or . GET ACKNOWLEDGEMENT

The object name (for which acknowledgement has been received) and the senders office is displayed on the screen. Automatically the ACKNOWLEDGEMENT field in the Mail log is updated to YES for the corresponding tuple.

4.7.2.3 Broadcasting Mode:

This mode is used when one wants to send short notices or messages to all offices simultaneously.

. MAIL NOTICE TO ALL

will perform this activity. The text of the message/notice has to be entered interactively. This function comes in handy if it is necessary to summon a meeting or a conference of personnel from all offices. The message/notice is displayed on the screen the next time one uses the OA system.

4.7.3 Mail Box Management:

The management of Mail box is an important activity that goes along with the Mailing and Despatching functions.

When one is examining the contents of one's Mail Box, the other offices should be denied access to that particular Mail box, for the time being. This is made possible by a switch whose two positions are OPEN and LOCK. When the MAIL

function is invoked the switch in the receiver's area is checked for OPEN position. If not the prospective sender is informed that the receiver's office is busy and that another attempt has to be made.

. GET MAIL BOX

thus turns the switch to LOCK position and sets it to OPEN position only after the activity is over.

4.7.4 Tracing:

When one has apriori knowledge of what offices an object has to pass through to get the final approval, the TRACE function becomes useful.

. TRACE <object name>

Starting with the first office (where the object originated), the function gives the names of other offices in sequence, to which the object has gone. If the object is stagnant in a particular office, a reminder could be sent to that office to expedite matters.

Of course, all along its path the object should bear the same name. This is important, so that the object does not get mixed up with other object(s) that would be exchanged between the other offices.

4.7.5 Noting:

Notings may be made in a letter or a particular document in a file

- . NOTE IN LETTER <NAME>
- . NOTE IN DOCNO <document number>
IN FILE <NAME>

The notings are attached immediately after the corresponding text. The number of notings on a letter/document is restricted to be one. However if a second noting is attached, it is overwritten on the previous note.

When an object is mailed to an office for its perusal and reply, the office may mail the reply to the sender. However, for its future reference, the office may wish to make a note in the object. This is done by calling the NOTE function.

The retrieval of notings is similar to the retrieval of documents.

4.7.6 Addresses of Other Offices:

Addresses of all offices connected by the OA system can be obtained by typing

- . GET ADRS or .GET ADDRESSES

The system then types the names and the addresses (PPN) of the other offices on the screen.

4.7.7 Creating Letters:

Letters may be created with a letter name by giving

. LETTER <NAME>

The text of the letter has to be keyed in and ended with FINIS . The letter is then created in the user's area.

4.7.8 Output Using TYPE/PRINT/READ:

The OFL allows information to be output in the following manner .

(i) Displayed on the screen

(ii) Sent to a line printer

(i) and (ii) may be achieved by TYPE OR PRINT respectively

. TYPE/PRINT <object>

READ command is used for reading letters/documents in another office for which the user may have access.

4.7.9 Office Text Processor:

An elementary text processor is available with the OA system for output purposes with an YES/NO option.

The processor has the following characteristics

. left margin 5 blanks

. width of text 80 characters

. lower or upper case letters (as it is keyed in)

4.7.10 Data-Base Queries:

The office maintains a data-base. It contains information about the various files/letters available in the office shelves. The information is maintained as a relation on which queries may be built.

The OA system accepts

- . conditions on more than one field
- . logical conditions

WHEN and DISPLAY statements are used successively for selection and projection.

WHEN and DELETE statements are used successively for deletion of tuple(s).

MODIFY is used to modify a tuple, while

INSERT is used to insert a tuple into the office data-base.

4.7.11 ON-LINE Assistance:

This facility is getting widespread importance with the advances in interactive computing. It gives general information rather than specific advise for the given situation. This is called the HELP facility in our OA system.

In our OA system, the user is required to type HELP at any point in time to get help. The system comes out with the reply about the various verbs available with the OA system and the corresponding activities they perform. The user may

try one or more verbs and there are Feedback and Error messages available for further support.

4.7.12 Feed-back and Error Messages:

Feed-back and Error messages flash on the screen whenever the user has committed a mistake. Something that makes the computer easy or hard to use is the way it talks to us. Good messages help make it easy for us to use; bad messages make it hard [3].

To design a program whose messages are relevant we have to commit ourselves to (i) being tolerant of user-errors, (ii) identifying messages people need. We should make use of what we know about people and how people think and feel when they use computers. People's expectation govern the way they react. Therefore, we should use familiar terms for the users to feel secure. The error messages have been developed with these in mind.

There are three classes of messages in our OA system.

(i) Error messages: which start with ;;;; gives a list of correct commands that go with the particular verb. This makes it easy for the user to diagnose his/her error.

(ii) Message: which starts with !!!!!. This is given when the activity is accomplished.

(iii) Prompt: This indicates that the system is waiting for an input from the user.

In addition to the above mentioned facilities some more elementary facilities are available (refer to Appendix C).

It may also be noted that whenever any information/message/notice is entered interactively, one has to end the session by typing 'FINIS' on a new line.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS:

The OA system developed can be efficiently used in an organisation where there exists information interchange between the various offices. The OA system is capable of mimicking the elementary office activities such as Mailing, Filing, Data-base querying, Letter processing, Letter/File tracing etc. (refer to Appendix C for demonstration).

The coding of our OA system software is in FORTRAN LANGUAGE. It has been sincerely tried to make the OA system a portable one. The program still contains some system dependent features. However, they can be modified quite easily by making appropriate changes in the DATA statements alone.

The text processor of the OA system is primitive but this shall not be considered as a negative point. This is because, nowadays, good text-processors are available on every conceivable system. Our OA system can be interfaced with one such text processor.

The limitations of our OA system lies in the fact that it has not been tested by the prospective users. So one really

does not know, how the users will receive the system as regards response time either to do an activity or to get an error message.

Another limitation is that Office Automation systems in general are considered to be cost displacement systems in the secretarial/clerical area. So it is necessary to allay the fears of the office workers before the system is implemented in any organisation.

5.2 RECOMMENDATIONS:

The following suggestions could be taken up for further work.

1. As a communication tool, synchronous messaging system may be provided to enable the offices to interrupt each other on their screens with short urgent messages. This would serve the purpose well rather than having to wait for the offices to use the OA system to know the notices or messages.
2. No archiving of letters/files is done at present. Thus a letter/file created long back (may not know be useful at all) will still be on the office shelves. The system should be provided with a facility to delete all these files/letters automatically and may be called back only when necessary.
3. Presently, when there is an error in the command given to the OA system it is necessary to retype the whole of the command again. It is possible that the user may commit an

error elsewhere in the command. The OA system may be designed so that the user need to retype only the incorrect portion of the command. This would definitely be to the advantage of the user.

4. Not many security features have been provided in our OA system. But these features can be incorporated in accordance with the specific requirements of each office.

REFERENCES

1. Aho, Alfred V. and Ullman, Jeffrey D., Principles of Compiler Design, Englewood Cliff, NJ, 1977.
2. Burns, Christopher, J., The Evolution of Office Information Systems, Datamation, April 1977, pp. 60-64.
3. Dean M., How a Computer Should Talk to People, IBM Systems Journal, Vol. 21, No. 4, 1982, pp.525-454.
4. Don Tapscott, Office Automation -- A User Driven Method, Plenum Press, 1982.
5. Hammer, Michael and Kunin, Jay S., OSL An Office Specification Language Description, OAM-024, Dec. 1980, MIT Lab. for CS.
6. Juliet, Sutherland and Marvin Sibru, Evaluation of an Office Analysis Methodology, OAM-043, Feb. 1983, MIT Lab. for CS.
7. Lee E. Heindel and Roberto Jerry, T., LANG-PAK -- An Interactive Language Design System, American Elsevier Publishing Inc., Prog. Language Series - 1, 1975.
8. Lochovsky Fred, H., Improving Office Productivity: A Technology Perspective, Proceedings of the IEEE, Vol.71, No. 4, April 1983, pp. 512-518.
9. Lum, V.Y., Choy, D.M., Shu, N.C., OPAS: An Office Procedure Automation System, IBM Systems Journal, Vol. 21, No. 3, 1982, pp. 327-351.
10. Michael Good, ETUDE and the Folklore of User Interface Design, OAM-030, March 1981, MIT Lab. for CS.
11. Sandra, L.Teger, Factors Impacting the Evolution of Office Automation, Proceedings of the IEEE, Vol. 71, No. 4, April 1983, pp. 512-518.
12. Traub, J.F., Quo Vadimus: Computer Science in a decade, Commun.ACM, Vol. 24, pp. 359, 1981.

13. Tsichritzis, D.C. and Lochovsky, Fred H., Data Models, Englewood Cliffs, NJ, Prentice Hall, 1982.
14. Vasista, T.S., Design and Implementation of an Automated Office Information System, M.Tech. Thesis, IIT Kanpur, June 1984.
15. Zloof, M.M., Office by Example: A Business Language that Unifies Data and Word-Processing and Electronic Mail, IBM Systems Journal, Vol. 21, No.3, 1982, pp. 272-306.

APPENDIX - A

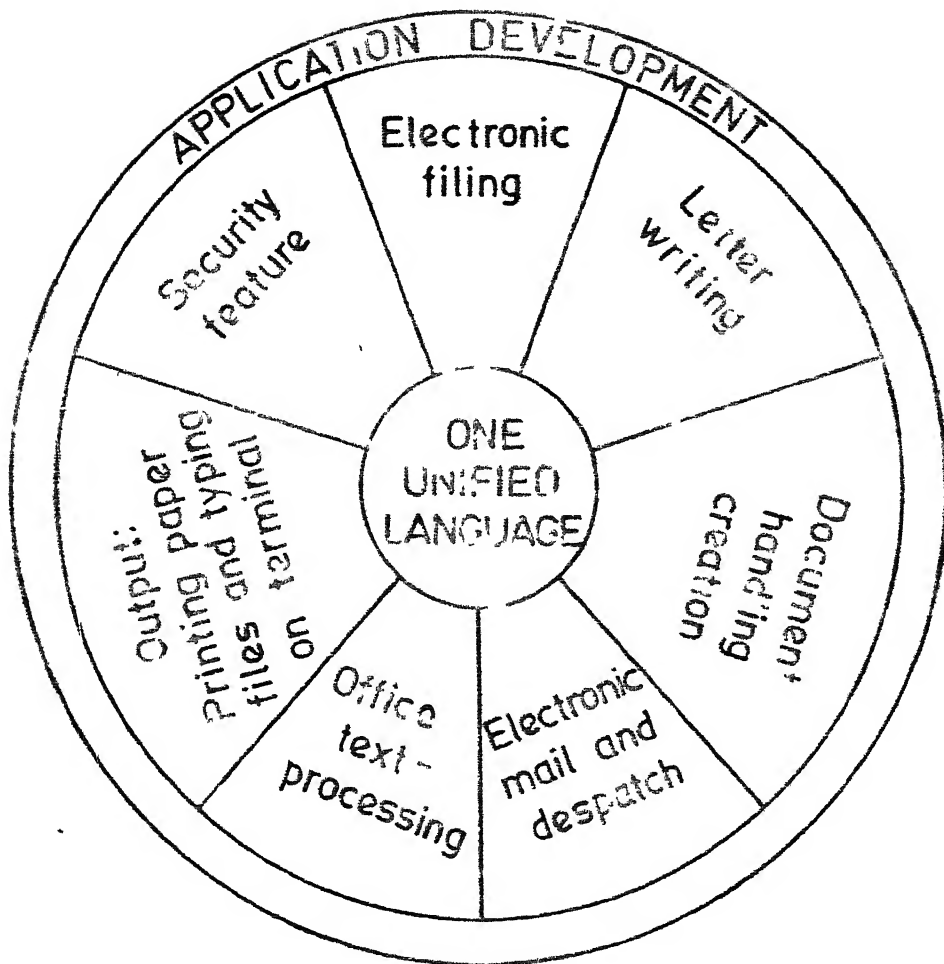


Fig. 1. Office automation system facilities.

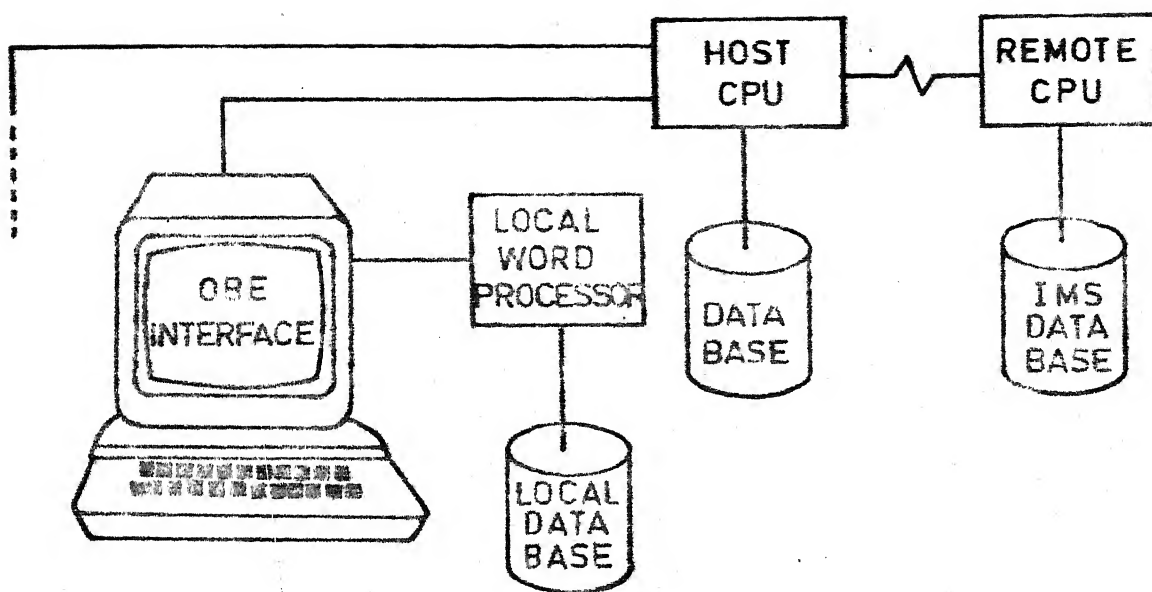


Fig. 2. System hardware configuration.

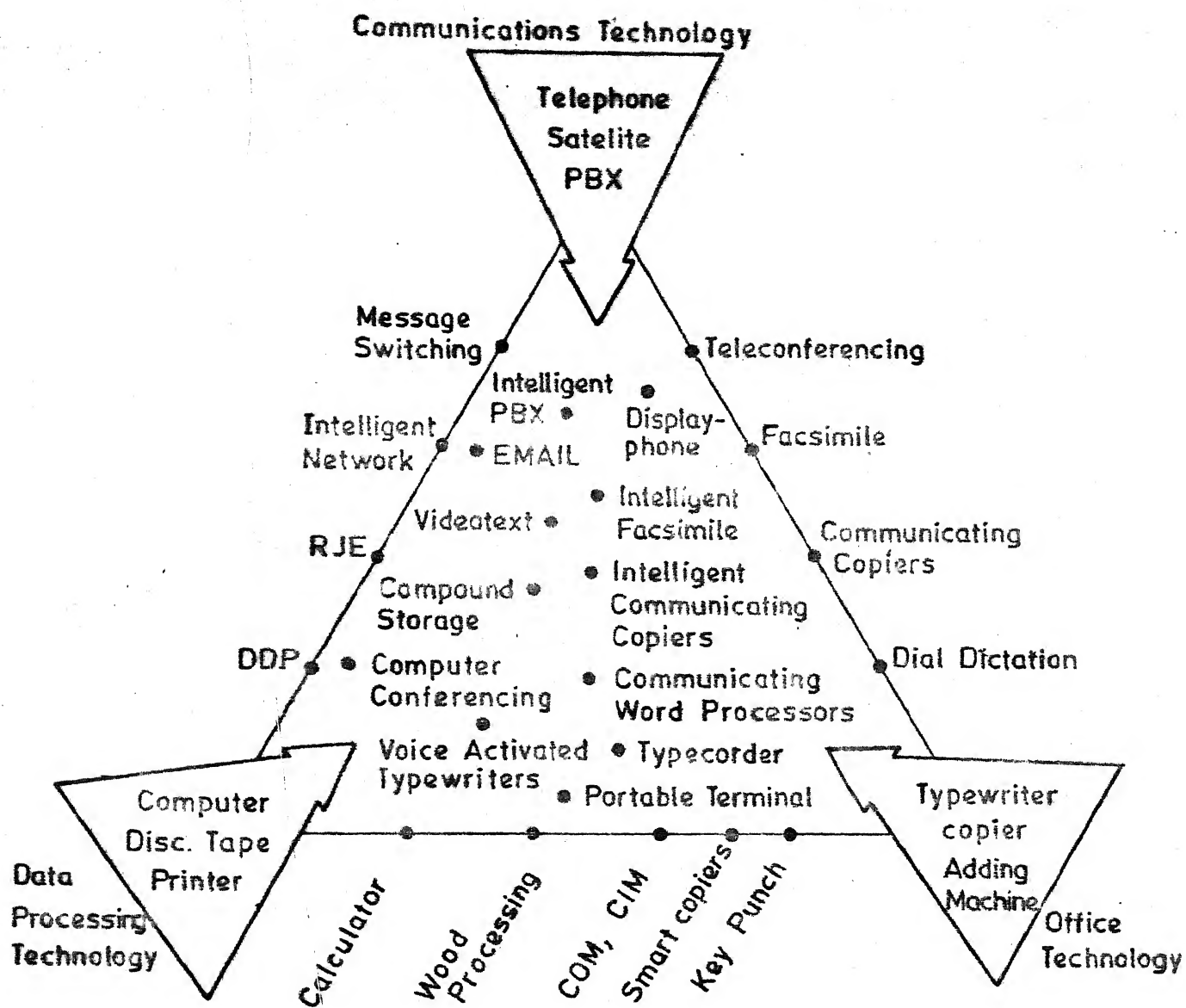


Fig. 3. Convergence of technologies in the office.

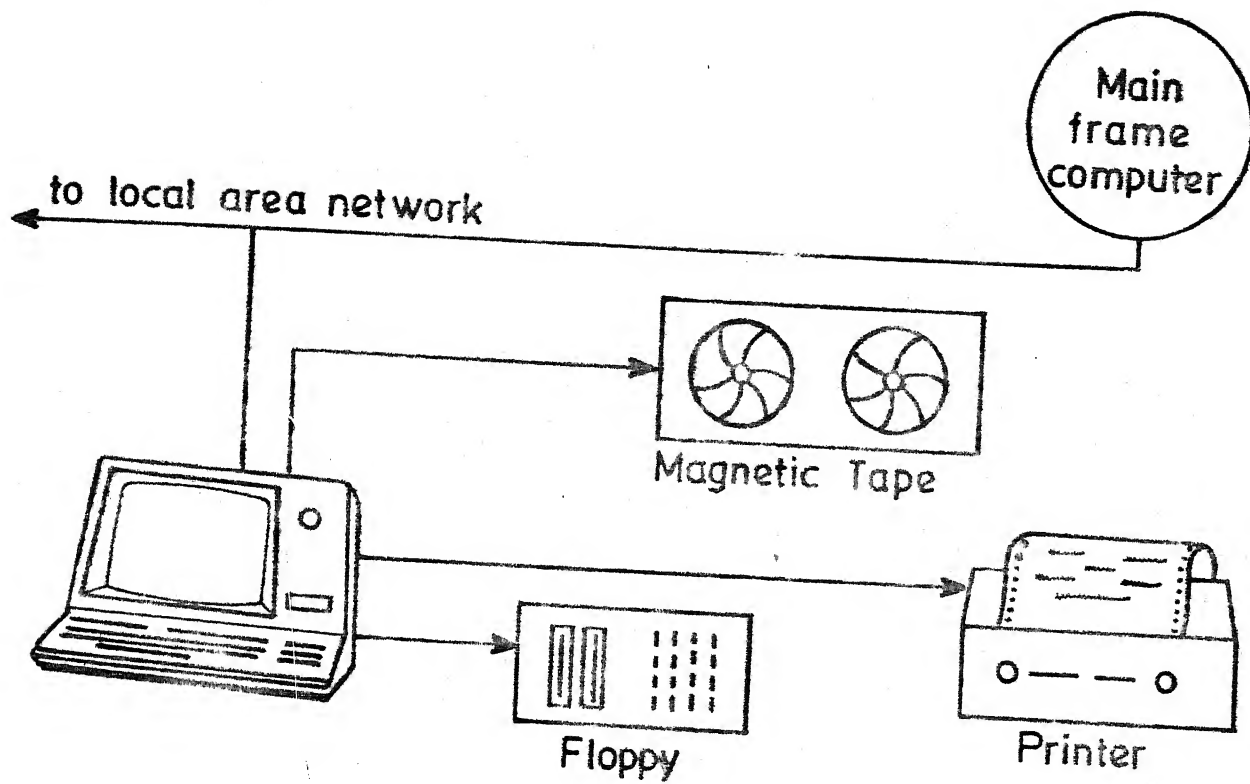


Fig. 4 A typical workstation layout.

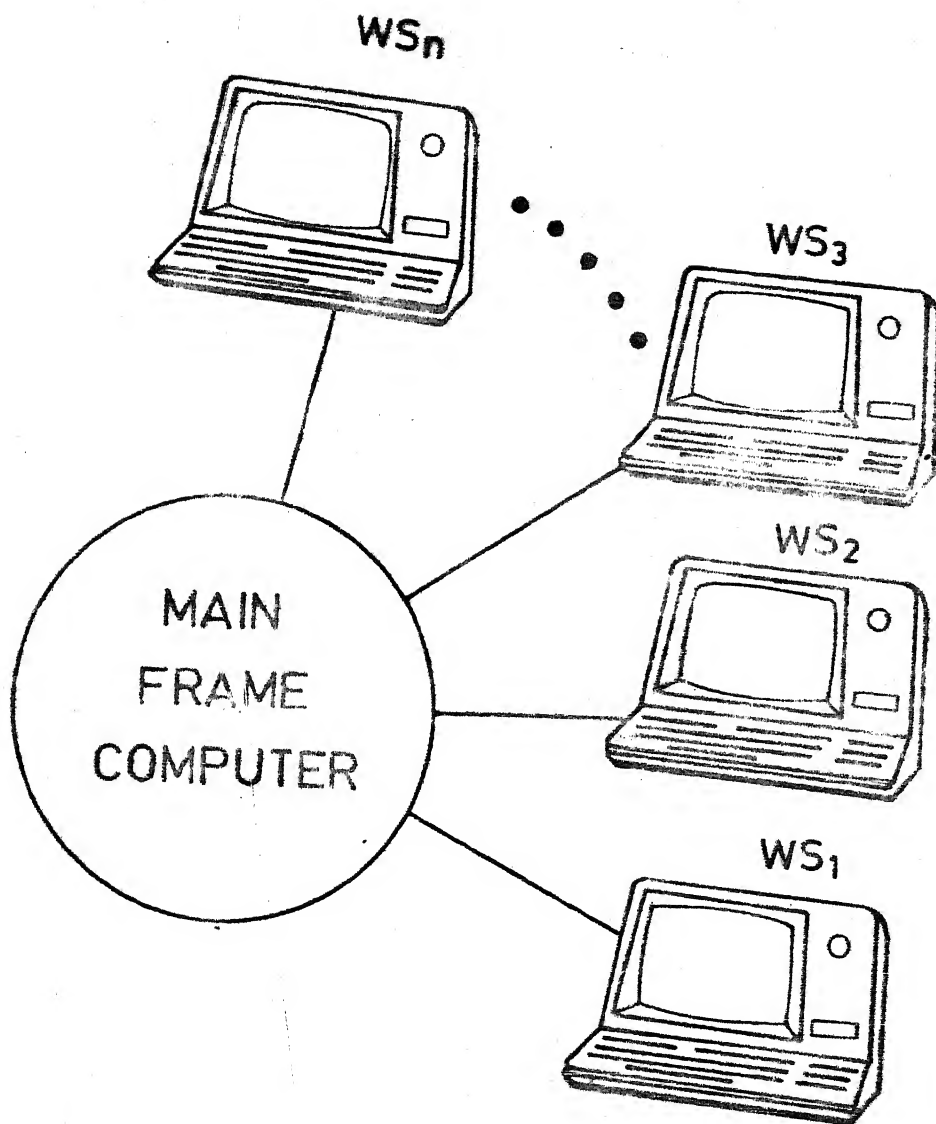


Fig. 5. Office workstation.

APPENDIX B

page 1

GRAMMAR OF THE "OFFICE LANGUAGE" (OFL)

```
<LIST>=("LETTER" "L" (1) S("TO") "TO" <PPN>! "L" (1) S("TO") "TO" <PPN>! "NOTICE" "TO" "ALL" (2)!S<MLERR>)
<MLERR>="PRINT INVALID "MAIL" COMMAND ->GIVE LIKE THISEXAMPLE =>> MAIL LXXXX TO 10100,100112 OR MAILNOTICE TO ALL'
<LIST>=("LETTER" "L" (1) S!"L" (1) S!"IN" "DOC" "NO" "(" (2) <T> ")"!S<NLERR>)
<MLERR>="PRINT INVALID "NOTE" LISTING'
<LIST>=("FILE" "F" S("IN") "IN" <PPN>! "F" S("IN") "IN" <PPN>!<OLERR>)
<OLERR>="PRINT INVALID "OPEN" LISTING ->GIVE LIKE THISFILE FXXXX IN 10100,100112 OR FXXX IN 10100,100112'
<LIST>=("LETTER" "L" (1) S!"L" (1) S!"FILE" "F"(2) S!"F" (2) S!S<PLERR>)
<PLERR>="PRINT INVALID "PRINT" COMMAND'
<PERR>="PRINT INVALID ITEM TO BE FILED IN'
<O>=("L" (1) S("IN")!"F" (2) S("IN")!"INFO" (3)!S<PERR>)
<PPN>=(T " " I)EOS
<DEPR>="PRINT DOCUMENT NUMBER IS NEEDED FOR THE ITEM TO BE FILED IN'
<O>=("WITH" "DOC" "NO" S!S<DERR>)
<DDP>=("EO"(1)!"NE"(2)!"GT"(3)!"LT"(4)!"GE"(5)!"LE"(6)!"<="(6)!="<="(6)!=">="(5)!=">="(5)!"<="(4)!">"(3)!"#"(2)!="="(1)!"
S<RRRT>))
<ERR>="PRINT INVALID RELATIONAL OPERATOR'
<MLERR>="PRINT INVALID "READ" LISTING'
<LIST>=("LETTER" "L" (1) S("IN") "IN" <PPN>! "L" (1) S("IN") "IN" <PPN>! "NOTE" "IN" "LETTER" "L" (2)S("IN") "IN"
<PPN>! "DOC" "NO" "(" (3) <T> ")"!"THE" "NOTE" "IN" "DOC" "NO" "(" (4) <T> ")"!S<RLERR>)
<LIST>=("LETTER" "L" (1) S!"L" (1) S!"FILE" "F" (2) S!"F" (2) S!S<TERR>),
<TERR>="PRINT INVALID "TYPE" LISTING: GIVE ::TYPE <FILE>OR TYPE <LETTER>'
<T>=<X> (" "<X>) (0 999)
<MLERR>="PRINT INVALID COPYING IN A FILE: GIVE IN FXXXXXX'
<U>=("IN" "F" S!S<UERR>)
<VALUE>=(I!S("OR" "AND" "&"))
<X>=(S(" " " "))
<ZERR>="PRINT THE REQUESTED ITEM FOR MODIFICATION IS IGNORED'
<Z>=("DATA" "BASE"! "D" "B"! "F" S<H>!S<ZERR>)
<AA>="PRINT INVALID "DELETE"'
<CLIST>=("FILE" "F" (1) S!"F" (1) S!"OFFICE" (2)!S<CLERR>)
<CLERR>="PRINT INVALID "CLOSE" LISTING :GIVE LIKE THIS CLOSE OFFICE OR CLOSE FILE FXXXX OR CLOSE FXXXX'
<CHOICE>=("OK" (0)!"AND" (1)!S<CEPR>)
```

```

<CLERR>='PRINT INVALID LOGICAL OPERATOR :YOU CAN USE "OR" OR "AND" ONLY'
<DO>=('EO"! "=")
<DELIST>=('LETTER' "L" (1) S!"L" (1) S!"FILE" "F" (2) S!"F" (2) S!"WHEN" "REC" "NO" (3) <DDD> I!S<AA>)
<DLIST>=('FILE' "F" S("TO") "TO" <PPN>!"F" S("TO") "TO" <PPN>!S<DLERR>)
<DESP>='PRINT INVALID DESPATCH LISTING :GIVE LIKE THIS DESPATCH FILE FXXXX OR FXXXX TO 10100,100112'
<FLIST>=('<P> "I" "F" S("WITH") <O>!S<FLERR>)
<FLERR>='PRINT INVALID FILE COMMAND :GIVE LIKE THIS FILE "LXXX OR FXXX OR INFO" IN FXXX WITH DOC NO IIT-005'
<FEPR>='PRINT INVALID FIELD NAME'
<FIELD>=('RECNO'(1 7 3)! "OF" (2 2 1)! "DT"(3 2 1)! "DR"(4 7 3)! "DS"(5 7 3)! "INFO"(6 2 1)! "DOC" "NO"(7 2 1)! "FILE"
(8 2 1)! "S"(",")<FEPR>!S("EO" "NE" "GT" "LT" "GE" "LE" "=" "<" "<=" ">" ">=" "##" "=")<FERR>!S<FERR>)
<GLERR>='PRINT INVALID "GET" COMMAND'
<GLIST>=('ADDRESSES' (1)! "ADRS" (1)! "MAIL" "BOX" (2)! "ALL" "DOC" "NO" (3)! "COPY" "OF" "((" (4) <T> ")"<U>! "LOG" (5)!
"ACKNO" (6)! "ACKNOWLEDGMENT" (6)!S<GLERR>)
<H>='PRINT "FILE" CANNOT BE MODIFIED'
<ICERR>='<FIELD> <POP> "TEST T8=T4 INVALID USE OF RELATIONAL OPERATOR' <VALUE> "TEST T9=T5 INVALID USE OF VALUE FIELD'
<KEPR>='PRINT INVALID "KEYWORD"'
<KEYERR>=('DISPLAY' (1)! "WHEN" (2)! "FILE" (3)! "DESPATCH" (4)! "MAIL" (5)! "OPEN" (6)! "GET" (7)!
"PRINT" (8)! "CLOSE" (9)! "MODIFY" (10)! "DELETE" (11)! "INSERT" (12)! "TYPE" (13)! "LETTER" (14)!
"READ" (15)! "TRACE" (16)! "NOTE" (17)!S<KERR>)
<LIST>=('FIELD' (" " <FIELD>) (0 7)EOS

```


APPENDIX - C1

In the following pages we give a demonstration of the various activities our OFFICE AUTOMATION SYSTEM (OA System) is capable of performing. In total there are FIVE SESSIONS.

The first session deals with the FEED-BACK and ERROR MESSAGES, HELP FACILITY that are available with the system and other MESSAGES that may appear from time to time while performing a TASK using the OA System.

The sessions two and three deal with MAILING and DESPATCHING which are the two most important activities of our OA System. Other related activities like CHECKING THE MAIL BOX, CHECKING ACKNOWLEDGMENT, READING, COPYING from a file, NOTING are also performed.

Session four introduces one to the FILING activity that is very common in most of the offices. Along with the FILING activity one gets to know how to retrieve documents, notings that may be present in the file.

Fifth session is the building of queries on DATA-BASE (of the office) which contains information on the various objects on the office shelves.

SO ON TO HAVE THE SESSIONS. GOOD LUCK.

VARAYANAN.

D F F I C E

password:

FILED

```

=====
4. FILEP: OFFICE AUTOMATION SYSTEM --OA SYSTEM.....
=====
for all verbs: one should input all the characters to use them.
=====
VERB                                DETAILS
=====
1. 43DP                             when it is given, types this page.
2. 0PST                             opens a file in a specified office
    >OPEN <file> in <PPN>
3. 00SE                             closes a OPENed file
    >CLOSE <file>
4. 03ET                             removes a file or a letter from
    office shelves
    >DELETE <file>
    >DELETE <letter>
5. 0ETTER                           prepares a letter
    >LETTER <letter-name>
6. FILE                             files objects in a specified file:
    >FILE INFO IN <file> WITH <docno>
    >FILE LETTER<letter> IN <file>
    WITH <docno>
    >FILE <file> IN FILE<file> WITH
    WITH <docno>
7. 0ET                             to get addresses
    >GET ADRS (or) GET ADDRESSES
    to receive Mail Box
    >GET MAIL BOX
    to get all docno in a file
    >GET ALL DOCNO
    (before giving this command,
=====

```

page 2

make sure the specified file is opened)
to get copy of certain docno into another file
>GET COPY OF (<docno1,...,docnoN>)
IN FILE<file>
{before giving this command, make sure the specified file is opened}
to know what acknowledgments have been received
>SET ACKNO OR GET ACKNOWLEDGMENT
to get information about objects sent to other offices --to where? whether acknowledgment received or not.
SET LOG
to send letter to the destination
>MAIL <letter> TO <destination>
to put our OA system in a broadcasting mode, in other words to send notices/ messages to all offices in the network
>MAIL NOTICE TO ALL
to send a file to the destination
>DESPATCH <file> TO <destination>
puts the mode for MODIFYING the OFFICE-data base.
>MODIFY
to append an entry in OFFICE-db.
to remove an entry in OFFICE-db.
>DELETE WHEN RECNO EQ 7
to type an object on the terminal
>TYPE LETTER<letter> !that exists
>TYPE FILE <file>! in our office!
to queue up an object to printer
>PRINT LETTER<letter>
>PRINT FILE <file>
to select the information in the OFFICE-db
{assuming the condition will be given next to this}
>WHEN FILE EQ FACADEMIC AND
OR LT 010184
to project the info on the domains of the OFFICE-db
>DISPLAY RECNO,DR,DS,FILE,INFO,DOCNO
{"DISPLAY" can be given before "WHEN" also}
to read on the terminal either a letter or a note in letter or a particular document in a file or a note in a document existing in our office or other offices.
>READ LETTER/NOTE IN LETTER/
DOC NO(...) /NOTE IN DOCNO (...) IN <PPM>.
to trace the whereabouts of an object (letter/file) sent from our office

8. WAIT

9. DISPATCH

10. MODIFY

11. TYPE

12. PRINT

13. WHEN

14. DISPLAY

15. READ

16. TRACE

17. NOTE

>TRACE <object>. to make notings in a letter/document
>NOTE IN <letter>/doc no(...)

OFFICE REQUEST:
>GET ADRS

GET ADRS

```
=====
*****
* ADDRESSES OF ALL OTHER OFFICES IN YOUR ORGANIZATION. *
*****
* OFFICE: DIRECTOR(IITK) ADDRESS: 1,1 *
* OFFICE: REGISTRAR(ACAD) ADDRESS: 1,2 *
* OFFICE: INR ADDRESS: 10100,100112 *
* OFFICE: AGRS ADDRESS: 16100,160070 *
* OFFICE: COMP SC. ADDRESS: 5100,50371 *
* OFFICE: MATHS. ADDRESS: 12100,120113 *
* OFFICE: MECH. ENGG. ADDRESS: 10100,100106 *
* OFFICE: MET. ENGG. ADDRESS: 14100,140070 *
* OFFICE: ADDRESS: *
=====
```

OFFICE REQUEST:
>READ DOC NO (IIT-005)

READ DOC NO (IIT-005)
!!!!!! This request is not recognized, NO FILE OPENed
please OPEN the <file> before this request.

OFFICE REQUEST:
>TRACE LACADEMIC

TRACE LACADEMIC
!!!!!!file LACADE.MIC does not exist in the mail log book of the office ME

OFFICE REQUEST:
>OPEN FILE FACADEMIC

OPEN FILE FACADEMIC
after OPEN is not recognized
please give like this:
>>OPEN FILE FYYY IN <OPN> (OR)
>>OPEN FYYY IN <OPN>

OFFICE REQUEST:
>DESPATCH LACADEMIC TO 10100,100106

DESPATCH LACADEMIC TO 10100,100106
LACADEMIC TO 10100,100106
INVALID DESPATCH LISTING :GIVE LIKE THIS DESPATCH FILE FXXXX OR FXXXX TO 10100.1

```

:::request not recognized:::
please give like this:
DISPATCH <file> TO 10100,100112 (or)
DTS <file> TO 10100,100112

```

```

OFFICE REQUEST:
>FILE INFORMATION IN FACADEMIC WITH DOC NO ME-003

```

```

FILE INFORMATION IN FACADEMIC WITH DOC NO ME-003
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
after FILE not recognized
pl. give like this
FILE LXXX IN FZZZ WITH DOCNO TIT-007 (OR)
FILE PXXX IN FZZZ WITH DOCNO RIT-007 (OR)
FILE INFO IN FZZZ WITH DOCNO GIT-007

```

```

OFFICE REQUEST:
>DISPLAY OF,OF

```

```

DISPLAY OF,OF
WHEN STATEMENT REQUIRED

```

```

OFFICE REQUEST:
>WHEN REQD=LINE

```

```

WHEN REQD=LINE
INVALID USE OF VALUE FIELD
WHEN STATEMENT REQUIRED

```

```

OFFICE REQUEST:
>LETTER LCOURSMEC

```

```

LETTER LCOURSMEC
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
please type the information that is to be FILED :
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The mechanical engineering students are required to do a minimum
course work of 24 credit hours and satisfy the requirement of
48 credit hours for an M.Tech degree by doing thesis.
DPGC CONVENER.

```

FINIS

request done! filed in:LCOURS.MEC

```

OFFICE REQUEST:
>MAIL LCOURSMEC TO 10100,100112

```

```

MAIL LCOURSMEC TO 10100,100112
!!!!request ignored!!!BUSY! pl. try some other time!

```

```

OFFICE REQUEST:
>READ NOTE IN LETTER LCOURSMEC

```

```

READ NOTE IN LETTER LCOURSMEC
::: after read not recognised:::
please give like this
read letter LXXX IN PPN OR
read LXXX IN PPN OR
read DOC NO (.....) OR
read NOTE IN LETTER LXXX IN PPN

```

read THE NOTE IN DOC NO (...,...)

OFFICE REQUEST:

>FILE FCOURSMEC IN FCOURSMEC WITH DOC NO COURS-1

FILE FCOURSMEC IN FCOURSMEC WITH DOC NO COURS-1

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 !!!request done! file:FCOURSMEC is FILED in file:FCOURSMEC

OFFICE REQUEST:

>OPEN FCOURSMEC IN 10100,100106

OPEN FCOURSMEC IN 10100,100106

!!!request done!!!file opened FILE:FCOURSMEC

OFFICE REQUEST:

>READ DOC NO (COURS-2,COURS-1)

READ DOC NO (COURS-2,COURS-1)

!!!!DOCUMENT: COURS-2 does not exist

*COURS-1 The mechanical engineering students are required to do a mi
 course work of 24 credit hours and satisfy the requirement
 credit hours for an M.Tech degree by doing thesis.
 DPGC CONVENER.

=====

DOCUMENT(READ):COURS-1

=====

!!!!request done!!!! DOCNOs read from FILE:FCOURSMEC

OFFICE REQUEST:

>NOTE IN DOC NO (COURS-1,COURS-2)

NOTE IN DOC NO (COURS-1,COURS-2)

;;;cannot make notings in
 more than one doc no: at a time;;;

OFFICE REQUEST:

>READ THE NOTE IN COURS-1

READ THE NOTE IN COURS-1

;;; after read not recognised;;;

please give like this

read letter LXXX IN PPN OR

read LXXX IN PPN OR

read DOC NO (...,...) OR

read NOTE IN LETTER LXXX IN PPN

read THE NOTE IN DOC NO (...,...) IN <PPN>

OFFICE REQUEST:

>OPEN FCOURSMEC IN 10100,100106

OPEN FCOURSMEC IN 10100,100106

!!!request done!!!file opened FILE:FCOURSMEC

OFFICE REQUEST:

>READ THE NOTE IN DOC NO (COURS-1)

READ THE NOTE IN DOC NO (COURS-1)

!!!! NOTE IN DOC NO: COURS-1 does not exist

!!!! request done!!!!

OFFICE REQUEST:
>CLOSE OFFICE

CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.

//////////for any information pl. call on NARAYANAN/hall-IV//////////

EXIT

SESSION TWO: MAILING AND DISPATCHING FUNCTIONS AND OTHER ACTIVITIES

date: 1

.RUI DAS

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100112

OFFICE
 =====

press today's date
 (dd-mm-yy):140685

identify your-self

office name:IME

password:

!!!!done!!!!

OFFICE_REQUEST:
 >LETTER LACADEMIC

LETTER LACADEMIC
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 please type the information that is to be FILEd :
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 THIS IS TO INFORM YOU THAT WE HAVE
 PROCESSED THE GRADES OF MECH.ENGG.
 STUDENTS DOING COURSES IN OUR DEPT.
 WE WILL BE SENDNG THE SAME IN A FEW
 DAYS.
 FINIS

request done! filed in:LACADE.MIC

OFFICE_REQUEST:
 >MAIL LACADEMIC TO 10100,100106

MAIL LACADEMIC TO 10100,100106
 !!!!request done!!! sent to the OFFICE!!!

OFFICE_REQUEST:
 >GET LOG

GET LOG

FROM	FILE/LETTER	TO	ACK
OFFICE!	NAME	OFFICE	RECD!
IME	LACADE.MIC	ME	NO

OFFICE_REQUEST:
 >GET ACKNO

GET ACKNO
 !!!NO ACKNOWLEDGEMENTS RECEIVED !!!

OFFICE REQUEST:

>GET MAIL BOX

GET MAIL BOX

!!!!NO MAIL RECEIVED !!!!

OFFICE REQUEST:

>FILE INED IN FACADEMIC WITH DOC NO IME-001

FILE INED IN FACADEMIC WITH DOC NO IME-001
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 please type the information that is to be FILED :
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 THE FOLLOWING ARE THE GRADES OF THE MECH.
 ENG. STUDENTS IN THE VARIOUS COURSES
 OFFERED BY THE IME DEPT.
 FINIS

request done! filed in:FACADE.MIC

OFFICE REQUEST:

>COPY FACADEMIC IN 10100,100112

JOB FACADEMIC IN 10100,100112
 !!!request done!!!file opened FILE:FACADE.MIC

OFFICE REQUEST:

>GET COPY OF (IME-001) IN FGRADEIME

GET COPY OF (IME-001) IN FGRADEIME
 =====
 DOCUMENT(COPIED):IME-001
 =====
 !!!!request done!!!!DOCNOs filed in FILE:FGRADE.IME

OFFICE REQUEST:

>DESPATCH FGRADEIME TO 10100,100106

DESPATCH FGRADEIME TO 10100,100106
 !!!!request done!!! sent to the OFFICE!!!

OFFICE REQUEST:

>GET LOG

GET LOG

FROM	FILE/LETTER	TO	ACK
OFFICE	NAME	OFFICE	RECD
IME	FACADE.MIC	ME	NO
IME	FGRADE.IME	ME	NO

OFFICE REQUEST:

>CLOSE OFFICE

CLOSE OFFICE

page 3

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.
EXIT :;;;;;;;;;for any information pl. call on VARAYANAN/hall-IV;;;;;;;;;;

THE FOLLOWING TRANSACTIONS TAKE PLACE IN ANOTHER OFFICE 10100,100106

.PVM DAS

O F F I C E
=====

press today's date
(dd-mmm-yy):140685

identify your-self

office name:ME

password:

!!!!done!!!!

OFFICE REQUEST:
>GET MAIL BOX

GET MAIL BOX

THE MAILBOX CONTAINS THIS MAIL:

LACADE.MIC FROM OFFICE: IME

THE MAILBOX CONTAINS THIS MAIL:

 UPGRADE TIME FROM OFFICE: TIME

OFFICE REQUEST:
 >MODIFY DATA BASE

MODIFY DATA BASE

you can MODIFY the office_db by using:
 INSERT: for inserting new data into office_db

 DELETE WHEN RECND = No.: for deleting that record.

 NOW !!ready!! FOR MODIFY

OFFICE REQUEST:
 >INSERT

INSERT please insert properly

OFF NO:(pl.give 3 digit #):001

Office From(OF):(pl.give 6 chr):IME

Office To(OT):(pl.give 6 chr):ME

Date Received(OR):(pl.give 6 digits):140684

Date Sent(OS):(pl.give 6 digits):140685

INFO:(pl.give info concisely; type and give
 carriage return last):GRADES OF MECH.STUDENTS

OFF NO:(pl.give in not more than 7 chr):ME-001

FILENAME:(pl.give in not more than 9 chr):LACADEIME

!!!!request done!!!office_db has been modified!!

OFFICE REQUEST:
 >MAIL NOTICE TO ALL

MAIL NOTICE TO ALL
 >please type in the notice to be sent to all the offices
 THERE WILL BE A MEETING OF ALL
 DEPARTMENTAL HEADS TO DISCUSS THE
 MODALITY OF SELECTION OF M.TECH
 STUDENTS AT 3.30P.M ON 15 JUNE 85.
 FINIS

(maximum 560 character)

!!!!request done!!!!
 notice sent to all offices

OFFICE REQUEST:
 >DELETE LACADEIME

page 5

DELETE LACADEIME
:::request ignored:::file:LACADE.IME doesnot exist on OFFICE shelves;

OFFICE_REQUEST:
>MODIFY DB

MODIFY DB

you can MODIFY the office_db by using:
INSERT: for inserting new data into office_db

DELETE WHEN RECNO = No.:for deleting that record.

NOW !!ready!! FOR MODIFY

OFFICE_REQUEST:
>INSERT

INSERT
please insert properly

REC NO:(pl.give 3 digit #):002
Office From(FR):(pl.give 6 chr):IME
Office To(OT):(pl.give 6 chr):ME
Date Received(DR):(pl.give 6 digits):140685
Date Sent(DS):(pl.give 6 digits):140685
INFO:(pl.give info concisely; type and give
carriage return last):GRADES OF MECH STUDENTS
DOC NO:(pl.give in not more than 7 chr):IME-001
Filename:(pl.give in not more than 9 chr):FGRADEIME

!!!!request done!!!office_db has been modified!!

OFFICE_REQUEST:
>GET LOG

GET LOG

FROM	FILE/LETTER	TO	ACK
OFFICE!	NAME	OFFICE	RECD!

OFFICE_REQUEST:
>GET ADRS

GET ADRS

=====

* ADDRESSES OF ALL OTHER OFFICES IN YOUR ORGANIZATION. *

page: 7

=====
note made in doc no: LME-001
=====
FACADE.LME

OFFICE-REQUEST:
>DESPATCH FACADE4IC TO 10100,100112

DESPATCH FACADEMIC TO 10100,100112
/////request ignored/////FILE:FACADE.MICdoes not exist in the OFFICE/////

OFFICE-REQUEST:
>DESPATCH FACADEMIC TO 10100,100112

DESPATCH FACADEMIC TO 10100,100112
FACADEMIC TO 10100,100112
INVALID DESPATCH LISTING :GIVE LIKE THIS DESPATCH FILE FXXXX OR FXXXX TO 10100.100
/////request not recognized/////

please give like this:
DESPATCH <file> TO 10100,100112 (or)
DES <file> TO 10100,100112

OFFICE-REQUEST:
>CLOSE OFFICE

CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.

EXIT ://///for any information pl. call on NARAYANAN/hall-IV/////

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100112

.RUV DAS
[:20:38]

OFFICE
=====

press today's date
(dd-mm-vy):150684

identify your-self

office name:IME

password:

!!!!!!000a!!!!!!

OFFICE FOR YOU READS:

THERE WILL BE A MEETING OF ALL DEPARTMENTIAL HEADS TO DISCUSS THE MODALIT
Y OF SELECTION OF M.TECH STUDENTS AT 3.30P.M. ON 15 JUNE 85.
SEND ON 14-06 FROM OFFICE ME

OFFICE REQUEST:

>GET LOG

GET LOG

FROM	FILE/LETTER	TO	ACK
OFFICE	NAME	OFFICE	RECD
IME	LACADE.MIC	ME	NO
IME	FGRADE.IME	ME	NO

OFFICE REQUEST:

>GET ACKNOWLEDGMENT

GET ACKNOWLEDGMENT

ACKNOWLEDGEMENT RECEIVED FOR THIS MAIL

LACADE.MIC FROM OFFICE: ME

ACKNOWLEDGEMENT RECEIVED FOR THIS MAIL

FGRADE.IME FROM OFFICE: ME

OFFICE REQUEST:

>GET LOG

GET LOG

FROM	FILE/LETTER	TO	ACK
OFFICE	NAME	OFFICE	RECD
IME	LACADE.MIC	ME	YES
IME	FGRADE.IME	ME	YES

OFFICE REQUEST:

>CLOSE OFFICE

page 9

CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.

EXIT

//////////for any information pl. call on NARAYANAN/hall-IV//////////

page 2

OFFICEL REQUEST:

>FILE LCOMPSMEC IN FCOMPSMEC WITH DOC NO COMP-01

FILE LCOMPSMEC IN FCOMPSMEC WITH DOC NO COMP-01

XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
!!!!!!request done!! file:LCOMPS.MEC is FILEd in file:FCOMPS.MEC

OFFICEL REQUEST:

>DELETE LCOMPSMEC

DELETE LCOMPSMEC

!!!!!!request done!! file deleted FILE:LCOMPS.MEC!!!!

OFFICEL REQUEST:

>DESPATCH FCOMPSMEC TO 10100,100112

DESPATCH FCOMPSMEC TO 10100,100112

!!!!!!request done!!! sent to the OFFICE!!!

OFFICEL REQUEST:

>CLOSE OFFICE

CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.

;;;;;;;;;;for any information pl. call on NARAYANAN/hall-IV;;;;;;;;;;

EXIT
.RUN JAS

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100112

OFFICE
=====

press today's date
(dd-mm-yy):160685

identify your-self

office name:IME

password:

!!!!done!!!!

OFFICELREQUEST:

>GET MAIL BOX

GET MAIL BOX

THE MAILBOX CONTAINS THIS MAIL:

FCOMPS.MEC FROM OFFICE: ME

OFFICELREQUEST:

>GET ACKNOWLEDGMENT

GET ACKNOWLEDGMENT

!!!!NO ACKNOWLEDGEMENTS RECEIVED !!!!

OFFICELREQUEST:

>OPEN FCOMPSMEC IN 10100,100112

OPEN FCOMPSMEC IN 10100,100112

!!!request done!!!file opened FILE:FCOMPS.MEC

OFFICELREQUEST:

>GET ALL DOC NO

GET ALL DOC NO

THE DOCUMENTS THAT ARE FILED IN FILE:FCOMPS.MEC
ARE AS FOLLOWS:

DOCUMENT NO:COMP-01

DOCUMENT NO:

!!!!request done!!!!the above are all the file has FILE:FCOMPS.MEC

OFFICELREQUEST:

>READ DOC NO (COMP-01)

READ DOC NO (COMP-01)

*COMP-01 TO, HEAD OF DEPT
IME

Dear Sir,

We wish to make use of the computing facility available in y
department for a couple of days.I request you to tell us the date and
time when we make use of the same.

With regards,

Head of Dept.
Mech.Engg.

=====

DOCUMENT(READ):COMP-01

=====

!!!!request done!!!! DOCNOs read from FILE:FCOMPS.MEC page 4

OFFICE REQUEST:
>LETTER LCOMPSIME

LETTER: LCDMP5IME
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
please type the information that is to be FILEd :
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
To,

XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
please type the information that is to be FILEd :.

XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
Fo,

Head of Dept.

Mech. Engrg.

Dear Sir, With reference to your file FCDWPSMEC sent to our department we wish to inform you that you can use the computing facility on 28th and 29th of June '85, between 20.30 HRS and 23.30 HRS.

With reference to your file FCDPMSEEC sent to our department to inform you that you can use the computing facility on 28th of June '85 between 20.30 HRS and 23.30 HRS.

With regards,

Head
IME.

FINI'S

request done! filed in: LCOMPS.IME

OFFICE REQUEST:

>MAIL LCOMPSIME TO 10100,100106

MAIL LCOMPSIME TO 10100,100106

!!!!!!request done!!!! sent to the OFFICE!!!!

OFFICE REQUEST:

>OPEN FCOMPSMEC IN 10100,100112

OPEN FCOMPSMEC IN 10100,100112

!!!request done!!!file opened FILE:FCOMPS.MEC

OFFICIAL REQUEST:

>NOTE IN DDC NO (COMP-01)

NOTE IN DOC NO (CDMP-01)

please type the text of the NOTE

22

The Mech. engg. department has been allowed to use our computing facility between 20.30 HRS and 23.30 HRS on 28th and 29th of June '85.

FINIS

```

=====
note made in doc no: COMP-01
=====
FCOMPS.MEC

```

OFFICIAL REQUEST:

>CLOSED OFFICE

CLOSE OFFICE

PAGE 5

 USE THIS OFFICE FOR FASTER COMMUNICATION
 USE THIS OFFICE FOR BETTER DECISION-MAKING
 USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.

thank you.
 ::::::::::for any information pl. call on NARAYANAN/hall-IV:::::::::

EXIT
RUN DAS

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100105

OFFICE
=====

press today's date
(dd-mm-yy):160685

identify your-self

office name: ME

password:

!!!!done!!!!

OFFICE REQUEST:

>FILE INFO IN FDPGCSMEC WITH DOC NO DPGC-01

FILE INFO IN FDPGCSMEC WITH DOC NO DPGC-01

FILE INFO IN FDPCSMC WITH DOC NO DPSC-01
XXXXXXXXXXXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXVXXXXXXXXXV
please type the information that is to be FILED :
XXXXXXXXXXXXXXXXXXXX

please type the information that is to be FILED :
XXXXXXXXXXVXXXXXXXXXVXXXXXXXXXVXXXXXXXXXVXXXXXXXXXVXXXXXXXXXVXXXXXXXXXVXXXXXXXXXV

There will be a meeting of all DPGC conveners to discuss the guidelines for grant of extension for M.Tech theses.

FINIS

request done! filed in:FDPGCS.MEC

OFFICE REQUEST:

>DESPATCH FDPGCSMEC. TO 10100,100112

DESPATCH FDPGCSMEC TO 10100,100112

DESPACH FORGOSMEL TO 10100,100112
!!!!!!request done!!!! sent to the OFFICE!!!!

OFFICIAL REQUEST:

>NOTE! IN DOC NO (DPGC-01)

NOTE: IN DOC NO (DPGC-01)

request is not recognised;
please open the file before this request;

OFFICER REQUEST:
>OPEN FDPGCSMEC IN 10100,100106

OPEN FDPGCSMEC IN 10100,100106
!!!request done!!!file opened FILE:FDPGCS.MEC

OFFICER REQUEST:
>NOTE IN DOC NO (DPGC-01)

NOTE IN DOC NO (DPGC-01)
please type the text of the NOTE
>>

In the meeting it was decided to constitute a committee which will
take care of granting extension to M.Tech theses.
FINIS

=====

note made in doc no: DPGC-01

=====

FDPGCS.MEC

OFFICER REQUEST:
>OPEN FCOMPSMEC IN 10100,100112

OPEN FCOMPSMEC IN 10100,100112
!!!request done!!!file opened FILE:FCOMPS.MEC

OFFICER REQUEST:
>GET ALL DOC NO

GET ALL DOC NO

THE DOCUMENTS THAT ARE FILED IN FILE:FCOMPS.MEC
ARE AS FOLLOWS:

DOCUMENT NO:COMP-01

DOCUMENT NO:

!!!!request done!!!!the above are all the file has FILE:FCOMPS.MEC

OFFICER REQUEST:
>READ THE NOTE IN DOC NO (COMP-01)

READ THE NOTE IN DOC NO (COMP-01)
COMP-01 The Mech. engg. department has been allowed to use our computing facility
between 20.30 HRS and 23.30 HRS on 28th and 29th of June '85.

=====

NOTE READ IN DOC NO: COMP-01

=====

!!!! request done !!!!!

OFFICER REQUEST:
>GET MAIL BOX

GET MAIL BOX

page 7

THE MAILBOX CONTAINS THIS MAIL:

LCDMPS.IME FROM OFFICE: IME

OFFICE REQUEST:
>GET ACKNOWLEDGMENT

GET ACKNOWLEDGMENT

ACKNOWLEDGEMENT RECEIVED FOR THIS MAIL

FCDMPS.MEC FROM OFFICE: IME

OFFICE REQUEST:
>TYPE LCDMPSIME
TYPE LCDMPSIME
DOCUMENT NO:

for this document;
do you need Office_txt_prc(Y/N):N

To,

Head of Dept.

Mech.Engg.

Dear Sir,

With reference to your file FCDMPSMEC sent to our department
we wish to inform you that you can use the computing facility on 28th
and 29th of June '85 between 20.30 HRS and 23.30 HRS.

With regards,

Head
IME.

!!!request done!!! PRINTED/TYPED the FILE:LCDMPS.IME

OFFICE REQUEST:
>CLOSE OFFICE
CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION

page 8
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.
EXIT ///////////////for any information pl. call on NARAYANAN/hall-IV//////////
.RUN DAS
THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100112

O F F I C E
=====

press today's date
(dd-mmm-yy):160685

identify your-self

office name:IME

password:

!!!!done!!!!

OFFICELREQUEST:
>GET MAIL BOX

GET MAIL BOX

THE MAILED BOX CONTAINS THIS MAIL:

FDPGCS.MEC FROM OFFICE: ME

OFFICELREQUEST:
>OPEN FDPGCSMEC IN 10100,100106

OPEN FDPGCSMEC IN 10100,100106
!!!request done!!!file opened FILE:FDPGCS.MEC

OFFICELREQUEST:
>READ THE NOTE IN DOC NO (DPGC-01)

READ THE NOTE IN DOC NO (DPGC-01)
DPGC-01 In the meeting it was decided to constitute a committee which will take
care of granting extension to M.Tech theses.

=====

NOTE READ IN DOC NO: DPGC-01

=====

!!!! request done !!!!!

OFFICER REQUEST:
>GET ACKNOW

GET ACKNOW

ACKNOWLEDGEMENT RECEIVED FOR THIS MAIL

LCDMPS.IME FROM OFFICE: ME.

OFFICER REQUEST:
>GET LOG

GET LOG

FROM	FILE/LETTER	TO	ACK
OFFICE	NAME	OFFICE	RECD
IME	LCDMPS.IME	ME	YES

OFFICER REQUEST:
>CLOSE OFFICE

CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.

EXIT :;;;;;;;;;for any information pl. call on NARAYANAN/hall-IV;;;;;;;;;;

SESSION FOUR: FILING AND RETRIEVAL OF DOCUMENTS

page 1

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100112

.RUN OAS
[9:11:38]

OFFICE
=====

press today's date
(dd-mmm-yy):230885

identify your-self

office name:IME

password:

!!!!done!!!!

OFFICE_REQUEST:
>LETTER LMONEYIME

LETTER LMONEYIME
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
please type the information that is to be FILEd :
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
To

The Director
IIT KANPUR

Dear Sir,
We wish to purchase an UPTRON D-10 Data Entry machine
We request you to grant funds for the same.

Sincerely yours
HEAD IME.

FINIS

request done! filed in:LMONEY.IME

OFFICE_REQUEST:
>MAIL LMONEYIME TO 10100,100130

MAIL LMONEYIME TO 10100,100130
!!!!request done!!! sent to the OFFICE!!!

OFFICE_REQUEST:
>FILE LMONEYIME IN FMONEYIME WITH DOC NO MONEY-001

FILE LMONEYIME IN FMONEYIME WITH DOC NO MONEY-001
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
!!!!request done! file:LMONEY.IME is FILEd in file:FMONEY.IME

OFFICE_REQUEST:

>CLOSE OFFICE
CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.
;;;;;;;;;for any information pl. call on NARAYANAN/hall-IV;;;;;;;;;

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100130

.RUN QAS
[9:14:33]

O F F I C E
=====

press today's date
(dd-mmm-yy):230885
identify your-self

office name:ME
password:

!!!!done!!!!

OFFICE_REQUEST:
>GET MAIL BOX
GET MAIL BOX

THE MAILED BOX CONTAINS THIS MAIL:

GET ACKNOW

OFFICE REQUEST:

READ LETTER LMONEYIME IN 10100,100130
THE LETTER BEING LOCATED IS :LMONEY.IME

Dear Sir

Sincerely yours
HEAD IME.

OFFICE REQUEST:

NOTE LETTER LMONEYIME

!!!! note attached to LETTER : LMONEY.IME

OFFICE REQUEST:

>LETTER LMONEYDIR

LETTER LMONEYDIR

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 please type the information that is to be FILED :

XXXXXXXXXXVXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV

XXXXXXX
To
Head IME

Dear Sir, We would request you to give more details about the machine D-10 A and the associated costs.

sincerely yours
DIRECTOR

FINIS

request done! filed in:LMONEY.DIR

OFFICE REQUEST:

>MAIL LMONEYDIR TO 10100,100112

MAIL LMONEYDIR TO 10100,100112

!!!!!!request done!!! sent to the OFFICE!!!!

OFFICE_REQUEST:
>CLOSE OFFICE
CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.
;;;;;;;;;for any information pl. call on NARAYANAN/hall-IV;;;;;;;;;

THE FOLLOWING TRANSACTIONS TAKE PLACE IN 10100,100112

.RUN OAS
[9:44:27]

O F F I C E
=====

press today's date
(dd-mmm-yy):240485

identify your-self

office name:IME

password:

!!!!done!!!!

OFFICE_REQUEST:
>GET ACKNOWLEDGMENT

GET ACKNOWLEDGMENT

ACKNOWLEDGEMENT RECEIVED FOR THIS MAIL

 LMONEY.IME FROM OFFICE: ME

OFFICE_REQUEST:
 >GET MAIL BOX

GET MAIL BOX

 THE MAILED BOX CONTAINS THIS MAIL:

 LMONEY.DIR FROM OFFICE: ME

OFFICE_REQUEST:
 >READ NOTE IN LETTER LMONEYIME IN 10100,100130

READ NOTE IN LETTER LMONEYIME IN 10100,100130
 THE LETTER BEING LOCATED IS :LMONEY.IME
 IN-ADEQUATE DETAILS REQUESTED THEM TO SEND A DETAILED LETTER
 WITH DETAILS AND THE ASSOCIATED COSTS.
 =====
 !!!!! note read in letter :LMONEY.IME
 =====

OFFICE_REQUEST:
 >READ LETTER LMONEYDIR IN 10100,100112

READ LETTER LMONEYDIR IN 10100,100112
 THE LETTER BEING LOCATED IS :LMONEY.DIR

To
 Head IME

Dear Sir,
 We would request you to give more details about the machine
 D-10 A and the associated costs.

sincerely yours
 DIRECTOR

=====

!!!!!! letter LMONEY.DIR read successfully

OFFICE_REQUEST:
 >FILE LMONEYDIR IN FMONEYIME WITH DOC NO MONEY-002

FILE LMONEYDIR IN FMONEYIME WITH DOC NO MONEY-002
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 !!!!!request done! file:LMONEY.DIR is FILED in file:FMONEY.IME

OFFICE_REQUEST:
 >NOTE IN DOC NO (MONEY-001)

NOTE IN DOC NO (MONEY-001)
 ;;;;request is not recognised;;;;;
 ;;;; please open the file before this request;;;;;

OFFICE_REQUEST:
 >OPEN FMONEYIME IN 10100,100112

OPEN FMONEYIME IN 10100,100112
 !!!!!request done!!!!file opened FILE:FMONEY.IME

page 6

OFFICE_REQUEST:
>NOTE IN DOC NO (MONEY-001)

NOTE IN DOC NO (MONEY-001)
please type the text of the NOTE
>>

A REQUEST FOR A DETAILED LETTER GIVING COSTS HAS BEEN
MADE BY THE DIRECTOR.
FINIS

=====

note made in doc no: MONEY-001

=====

FMONEY.IME

OFFICE_REQUEST:
>LETTER LMONEYIME

LETTER LMONEYIME
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
please type the information that is to be FILED :
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
TO

The Director
IIT KANPUR
Dear Sir,

In regard to the purchase of an educational software
package we request you to direct the ACCTS.Dept to issue us
a cheque for 250 dollars

Thanking you,

Sincerely
HEAD IME

FINIS

request done! filed in:LMONEY.IME

OFFICE_REQUEST:
>FILE LMONEYIME IN FMONEYIME WITH DOC NO MONEY-003

FILE LMONEYIME IN FMONEYIME WITH DOC NO MONEY-003
XXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXVXXXXXXXXXXV
!!!!request done! file:LMONEY.IME is FILED in file:FMONEY.IME

OFFICE_REQUEST:
>CLOSE OFFICE

CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

page 7

thank you.
:;;;;;;;;;for any information pl. call on NARAYANAN/hall-IV;;;;;;;;;;

SESSION FIVE:DATA-BASE QUERYING AND RETRIEVAL.

page 1

.RUN OAS

OFFICE
=====

press today's date
(dd-mmm-yy):200785

identify your-self

office name:ME

password:

!!!!done!!!!

OFFICE_REQUEST:
>WHEN RECNO=>1

WHEN RECNO=>1
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY RECNO,INFO,DR,DS,FILE,INFO,DOCNO

DISPLAY RECNO,INFO,DR,DS,FILE,INFO,DOCNO

REC NO	OFFICE FROM	OFFICE TO	DATE RECVD	DATE SENT	INFORMATION	DOC NO	FILE NAME
001!	!	!	160685!	150685!	INFO ON GRADES	IME-001!	FGRADES
004!	!	!	120585!	100485!	USAGE OF COMPUTER	ICHEN-02!	FCOMPUTER
006!	!	!	200685!	150685!	COMBINED COURSES	IME-003!	FCOURSIME
008!	!	!	170685!	140685!	PLACEMENT ACTIVITY	IME-004!	FPLACEMEN

OFFICE_REQUEST:
>WHEN FILE=FGRADES

WHEN FILE=FGRADES
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY INFO,OF,OT,DS,DR

DISPLAY INFO,OF,OT,DS,DR

REC NO	OFFICE FROM	OFFICE TO	DATE RECVD	DATE SENT	INFORMATION	DOC NO	FILE NAME
!	IME	!	160685!	150685!	INFO ON GRADES	!	!

OFFICE_REQUEST:
>WHEN RECNO >5 AND OF=IME

WHEN RECNO >5 AND OF=IME
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY RECNO,OF,FILE,INFO

DISPLAY RECNO,OF,FILE,INFO

RECNO	OFFICE	OFFICE	DATE	DATE	INFORMATION	DOC	FILE
NO	FROM	TO	RECD	SENT		NO	NAME
006	IME	!	!	!	COMBINED COURSES	!	FCOURSIME
008	IME	!	!	!	PLACEMENT ACTIVITY	!	FPLACEMEN

OFFICE_REQUEST:
>WHEN RECNO=IIT

WHEN RECNO=IIT
INVALID USE OF VALUE FIELD
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>WHEN DR=170685 OR DS=150685

WHEN DR=170685 OR DS=150685
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY DR,DS,RENO,OF,OT,FILE

DISPLAY DR,DS,RENO,OF,OT,FILE
RENO
INVALID FIELD NAME
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>WHEN FILE#FGRADES

WHEN FILE#FGRADES
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY FILE,INFO

DISPLAY FILE,INFO

RECNO	OFFICE	OFFICE	DATE	DATE	INFORMATION	DOC	FILE
NO	FROM	TO	RECD	SENT		NO	NAME
!	!	!	!	!	USAGE OF COMPUTER	!	FCOMPUTER
!	!	!	!	!	COMBINED COURSES	!	FCOURSIME

===== ! PLACEMENT ACTIVITY ! FPLACEMEN =====

OFFICE_REQUEST:
>WHEN FILE >FGRADES

WHEN FILE >FGRADES
INVALID USE OF RELATIONAL OPERATOR
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>WHEN RENO >1

WHEN RENO >1
RENO >1&
INVALID FIELD NAME
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>WHNE

WHNE
WHNE
INVALID "KEYWORD"

OFFICE_REQUEST:
>WHEN RECNO > 4 OR RECNO < 4

WHEN RECNO > 4 OR RECNO < 4
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY RECNO,FILE,OF,OT,INFO

DISPLAY RECNO,FILE,OF,OT,INFO

REC NO	OFFICE FROM	OFFICE TO	DATE RECVD	DATE SENT	INFORMATION	DOC NO	FILE NAME
001	IME	IME	!	!	!INFO ON GRADES	!	!FGRADES
006	IME	CS	!	!	!COMBINED COURSES	!	!FCOURSIME
008	IME	CIVIL	!	!	!PLACEMENT ACTIVITY	!	!FPLACEMEN

OFFICE_REQUEST:
>WHEN RECNO # 4

WHEN RECNO # 4
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY RECNO,FILE,OF,OT,INFO

DISPLAY RECNO,FILE,OF,OT,INFO

REC NO	OFFICE FROM	OFFICE TO	DATE RECVD	DATE SENT	INFORMATION	DOC NO	FILE NAME
-----------	----------------	--------------	---------------	--------------	-------------	-----------	--------------

```

01!IME !ME ! ! !INFO ON GRADES ! !FGRADES
=====
06!IME !CS ! ! !COMBINED COURSES ! !FCOURSIME
=====
08!IME !CIVIL ! ! !PLACEMENT ACTIVITY ! !FPLACEMEN
=====

```

OFFICE_REQUEST:
>WHEN DR=170685

WHEN DR=170685
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DIPLAY

DIPLAY
DIPLAY
INVALID "KEYWORD"

OFFICE_REQUEST:
>DISPLAY DR,DS,OF,OT

DISPLAY DR,DS,OF,OT

```

=====
REC|OFFICE|OFFICE| DATE | DATE | INFORMATION | DOC | FILE
NO |FROM | TO | RECVD| SENT | | NO | NAME
=====
!IME !CIVIL !170685!140685! ! !
=====

```

OFFICE_REQUEST:
>DELETE WHEN RECNO=9

DELETE WHEN RECNO=9
;;;;request ignored;;the record # 9 has not been found;;;

OFFICE_REQUEST:
>WHEN RECNO =>1

WHEN RECNO =>1
DISPLAY STATEMENT REQUIRED

OFFICE_REQUEST:
>DISPLAY RECNO,OF,OT,FILE,INFO,DOCNO

DISPLAY RECNO,OF,OT,FILE,INFO,DOCNO
*CH*U

```

=====
REC|OFFICE|OFFICE| DATE | DATE | INFORMATION | DOC | FILE
NO |FROM | TO | RECVD| SENT | | NO | NAME
=====
001!IME !ME ! ! !INFO ON GRADES !IME-001!FGRADES
=====
004!CHEM !CIVIL ! ! !USAGE OF COMPUTER !CHEM-02!FCOMPUTER
=====
006!IME !CS ! ! !COMBINED COURSES !IME-003!FCOURSIME
=====
008!IME !CIVIL ! ! !PLACEMENT ACTIVITY !IME-004!FPLACENEN
=====

```

page 5

OFFICE_REQUEST:
CLOSE OFFICE
CLOSE OFFICE

USE THIS OFFICE FOR FASTER COMMUNICATION
USE THIS OFFICE FOR BETTER DECISION-MAKING
USE THIS OFFICE FOR OFFICIAL PURPOSES

thank you.
;;;;;;;;;for any information pl. call on NARAYANAN/hall-IV;;;;;;;;;